



**ITS**

Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KS 141501**

# **OPTIMASI PENENTUAN RUTE KENDARAAN DISTRIBUSI PRODUK AIR MINUM KEMASAN GALON MENGGUNAKAN KOMBINASI ALGORITMA GENETIKA DAN PENCARIAN TABU DI DEPOT AIR MINUM ISI ULANG BANYU BELIK, PURWOKERTO**

NISA SETYA DINI  
NRP 5211 100 186

Dosen Pembimbing :  
Renny Pradina Kusumawardani, S.T., M.T.  
Amalia Utamima, S.Kom., MBA.

JURUSAN SISTEM INFORMASI  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KS 141501**

# **VEHICLE ROUTING OPTIMIZATION OF GALON PACKAGED WATER DISTRIBUTION USING THE COMBINATION OF GENETIC ALGORITHM AND TABU SEARCH AT WATER REFILL DEPOT IN BANYU BELIK, PURWOKERTO**

**NISA SETYA DINI  
NRP 5211 100 186**

**SUPERVISOR:**  
Renny Pradina Kusumawardani, S.T., M.T.  
Amalia Utamima, S.Kom., MBA.

**JURUSAN SISTEM INFORMASI  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015**

## LEMBAR PENGESAHAN

# **OPTIMASI PENENTUAN RUTE KENDARAAN DISTRIBUSI PRODUK AIR MINUM KEMASAN GALON MENGGUNAKAN KOMBINASI ALGORITMA GENETIKA DAN PENCARIAN TABU DI DEPOT AIR MINUM ISI ULANG BANYU BELIK, PURWOKERTO**

## **TUGAS AKHIR**

**Disusun Untuk Memenuhi Salah Satu Syarat**

**Memperoleh Gelar Sarjana Komputer**

**pada**

**Jurusan Sistem Informasi**

**Fakultas Teknologi Informasi**

**Institut Teknologi Sepuluh Nopember**

**Oleh:**

**NISA SETYA DINI**

**NRP. 5211 100 186**

**Gababaya, 2015**

**KETUA JURUSAN SISTEM INFORMASI**

**Dr. Eng. FEBRILIYAN SAMOPA, S.Kom., M.Kom.**

**NIP.19730219 199802 1 001**



## LEMBAR PERSETUJUAN

# **OPTIMASI PENENTUAN RUTE KENDARAAN DISTRIBUSI PRODUK AIR MINUM KEMASAN GALON MENGGUNAKAN KOMBINASI ALGORITMA GENETIKA DAN Pencarian Tabu DI DEPOT AIR MINUM ISI ULANG BANYU BELIK, PURWOKERTO**

## **TUGAS AKHIR**

**Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada**

**Jurusan Sistem Informasi**

**Fakultas Teknologi Informasi**

**Institut Teknologi Sepuluh Nopember**

**Oleh :**

**NISA SETYA DINI**

**NRP. 5211 100 186**

**Disetujui Tim Penguji : Tanggal Ujian : 6 Juli 2015**

**Periode Wisuda: September 2015**

**Renny Pradina Kusumawardani, S.T., M.T.**

**(Pembimbing I)**

**Amalia Utamima, S.Kom., MBA.**

**(Pembimbing II)**

**Wiwik Anggraeni, S.Si., M.Kom.**

**(Penguji I)**

**Irmasari Hafidz S.Kom., M.Sc.**

**(Penguji II)**

**OPTIMASI PENENTUAN RUTE KENDARAAN  
DISTRIBUSI PRODUK AIR MINUM KEMASAN  
GALON MENGGUNAKAN KOMBINASI  
ALGORITMA GENETIKA DAN PENCARIAN TABU  
DI DEPOT AIR MINUM ISI ULANG BANYU BELIK,  
PURWOKERTO**

**Nama Mahasiswa : Nisa Setya Dini**  
**NRP : 5211 100 186**  
**Jurusan : Sistem Informasi FTIf-ITS**  
**Pembimbing 1 : Renny Pradina K, S.T., M.T.**  
**Pembimbing 2 : Amalia Utamima, S.Kom., MBA.**

**ABSTRAK**

*Distribusi merupakan suatu proses memindahkan suatu barang dari pemasok hingga ke pelanggan akhir. Akan tetapi pada kenyataannya, proses pengiriman barang dari pemasok ke pelanggan kurang memperhatikan biaya yang dikeluarkan, karena tidak efisiennya rute pengiriman barang yang dilakukan. Masalah ini sering disebut dengan Vehicle Routing Problem (VRP).*

*Depot Air Minum Isi Ulang (Damiu) Banyu Belik merupakan sebuah depot yang sedang berkembang di Kabupaten Banyumas, tetapi belum memperhatikan rute pengiriman, hanya berdasarkan perkiraan dari pihak pengantar. Pengiriman dan pengambilan galon yang tidak memperhatikan rute optimal hanya akan menambah biaya pengiriman. Untuk meminimalkan biaya yang dikeluarkan dalam pengiriman dan pengambilan galon, perlu meminimalkan armada kendaraan dan jumlah jarak perjalanan, dengan batasan bahwa kendaraan harus memiliki kapasitas yang cukup untuk mengangkut barang yang akan dikirim dan mengambil barang pada pelanggan untuk mengembalikan barang ke depot.*

*Metode yang digunakan untuk optimasi rute pengiriman barang dalam studi kasus Depot Air Minum Banyu Belik adalah metode kombinasi algoritma genetika dan pencarian tabu. Algoritma genetika memiliki kelebihan memberikan kemungkinan solusi yang banyak sedangkan pencarian tabu menutupi kekurangan dari algoritma genetika yang kurang mampu dalam pengidentifikasian solusi lokal sehingga proses pencarian mengalami pengulangan pada daerah solusi yang sama.*

*Hasil uji coba menunjukkan bahwa penentuan rute menggunakan algoritma genetika dan pencarian tabu lebih efisien dibandingkan rute awal yang telah ditentukan Depot Banyu Belik.*

**Kata Kunci:** *Vehicle Routing Problem with Pick-up and Delivery, Rute, Optimasi, Algoritma Genetika, Pencarian Tabu.*

**VEHICLE ROUTING OPTIMIZATION OF GALON  
PACKAGED WATER DISTRIBUTION USING THE  
COMBINATION OF GENETIC ALGORITHM AND TABU  
SEARCH AT WATER REFILL DEPOT IN BANYU BELIK,  
PURWOKERTO**

**Student Name : Nisa Setya Dini**  
**NRP : 5211 100 186**  
**Department : Sistem Informasi FTIf-ITS**  
**Supervisor 1 : Renny Pradina K, S.T., M.T.**  
**Supervisor 2 : Amalia Utamima, S.Kom., MBA.**

**ABSTRACT**

*Distribution is a process of moving the goods from the supplier to customers. But in fact, the delivery of goods from supplier to customers less attention to the costs incurred, due to the inefficient delivery route is performed. This problem is often called the Vehicle Routing Problem (VRP).*

*Banyu Belik Water Refill Depot is a depot that is growing in Banyumas, but not attention to the delivery route, just based on estimates of the delivery man. Delivery and retrieval of galons that do not attention to the optimal route would only add to the cost of shipping. For minimum costs incurred in the delivery and retrieval galon, it is necessary to minimize the vehicle fleet and the amount of travel distance, with the restriction that the vehicle must have sufficient capacity to transport the goods to be shipped and take the goods to the customers to return the goods (empty galon) to the depot.*

*The method used for optimization of the delivery route in this observation is combination of genetic algorithm and tabu search. Genetic algorithms have advantages provide possible solutions to the many taboos while searching shortfall of genetic algorithms*

*that are less capable in identifying local solutions so the algorithm search process into a loop in the area of the same solution.*

*Experimental results show that the determination of the genetic algorithm and tabu search is more efficient than a predetermined route beginning from Banyu Belik Depot.*

**Keywords:** *Vehicle Routing Problem with Pick-up and Delivery, Route, Optimization, Genetic Algorithm, Taboo Search.*



## DAFTAR ISI

ABSTRAK .....	v
ABSTRACT .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang Masalah .....	1
1.2. Rumusan Masalah .....	4
1.3. Batasan Pengerjaan Tugas Akhir .....	5
1.4. Tujuan Penelitian .....	5
1.5. Manfaat Penelitian .....	5
BAB II DASAR TEORI .....	6
2.1. Depot Air Minum Isi Ulang Banyu Belik .....	6
2.2. Vehicle Routing Problem (VRP) .....	7
2.3. <i>Vehicle Routing Problem with Pick-up and Delivering</i> (VRPPD) .....	8
2.4. Algoritma Genetika .....	10
2.4.1. Proses Algoritma Genetika .....	11
2.5. Algoritma Pencarian Tabu .....	15
2.6. Kombinasi Algoritma Genetika dan Pencarian Tabu ..	18
2.7. Mekanisme Kombinasi Algoritma Genetika dan Pencarian Tabu .....	20
BAB III METODOLOGI Pengerjaan Tugas Akhir ...	23
3.1. Penyiapan Data .....	23
3.1.1. Pengumpulan Data .....	23
3.1.1. Praproses Data .....	24
3.2. Implementasi Metode .....	24
3.3. Validasi .....	26
3.4. Analisis Hasil .....	26

3.5.	Dokumentasi Tugas Akhir .....	26
BAB IV PERANCANGAN .....		27
4.1.	Deskripsi Data.....	27
4.2.	Desain Permodelan Permasalahan .....	30
4.3.	Desain Algoritma .....	34
4.3.1.	Desain Algoritma Genetika .....	35
4.3.2.	Desain Algoritma Pencarian Tabu.....	37
4.3.3.	<i>Hybrid GA-TS</i> .....	37
BAB V IMPLEMENTASI .....		43
5.1.	Implementasi Algoritma Genetika .....	43
5.1.1.	Inisialisasi Parameter.....	43
5.1.2.	Inisialisasi Populasi .....	44
5.1.3.	Melakukan Perhitungan Nilai Fitness dan Elitisme .....	45
5.1.4.	Seleksi .....	46
5.1.5.	Crossover.....	46
5.1.6.	Mutasi.....	48
5.2.	Kriteria Pemberhentian .....	49
5.3.	Implementasi Algoritma Pencarian Tabu.....	49
5.4.	Nilai Fitness Akhir .....	52
BAB VI HASIL DAN PEMBAHASAN.....		55
6.1.	Lingkungan Uji Coba.....	55
6.2.	Parameter Uji Coba.....	56
6.3.	Skenario Uji Coba.....	56
6.4.	Hasil Uji Coba Dataset.....	57
6.4.1.	Hasil Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300 , $P_m$ 0.05 , dan $P_c$ 0.2-0.9.....	57
6.4.2.	Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300 , $P_m$ 0.1 , dan $P_c$ 0.2-0.9 .....	59
6.4.3.	Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300, $P_c$ = 0.2-0.9, $P_m$ = 0.05, dan Kombinasi Algoritma Pencarian Tabu.....	61

6.4.4.	Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300, $P_c = 0.2-0.9$ , $P_m = 0.1$ , dan Kombinasi Algoritma Pencarian Tabu .....	63
6.5.	Hasil Uji Coba Data Perusahaan .....	64
6.5.1.	Hasil Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300, $P_m$ 0.05, dan $P_c$ 0.2-0.9 .....	65
6.5.2.	Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300, $P_m$ 0.1, dan $P_c$ 0.2-0.9 .....	67
6.5.3.	Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300, $P_c = 0.2-0.9$ , $P_m = 0.05$ , dan Kombinasi Algoritma Pencarian Tabu .....	69
6.5.4.	Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300, $P_c = 0.2-0.9$ , $P_m = 0.1$ , dan Kombinasi Algoritma Pencarian Tabu .....	72
6.5.5.	Bagan Skenario .....	74
6.6.	Analisa Hasil Uji Coba Dataset .....	75
6.6.1.	Analisa Hasil Uji Coba Algoritma Genetika .....	75
6.6.2.	Analisis Hasil Uji Coba Algoritma Genetika dan Pencarian Tabu .....	77
6.6.3.	Perbandingan Hasil Algoritma Genetika dengan Algoritma Genetika dan Pencarian Tabu .....	78
6.7.	Analisa Hasil Uji Coba Data Perusahaan .....	79
6.7.1.	Analisa Hasil Uji Coba Algoritma Genetika .....	79
6.7.2.	Analisis Hasil Uji Coba Kombinasi Algoritma Genetika dan Pencarian Tabu .....	80
6.7.3.	Perbandingan Hasil Algoritma Genetika dengan Algoritma Genetika dan Pencarian Tabu .....	82
6.8.	Validasi .....	83
6.9.	Perbandingan Rute Hasil GA-TS dengan Rute Awal ..	91
BAB VII KESIMPULAN DAN SARAN .....		101

7.1. Kesimpulan .....	101
7.2. Saran .....	102
DAFTAR PUSTAKA.....	103
BIODATA PENULIS.....	107



## DAFTAR TABEL

Tabel 4.1 Data Jarak (1) .....	A-38
Tabel 4.2 Data Jarak (2) .....	A-39
Tabel 6.1 Lingkungan Uji Coba Perangkat Keras .....	55
Tabel 6.2 Lingkungan Uji Coba Perangkat Lunak .....	55
Tabel 6.3 Uji Coba $N_{pop} = 100$ , $P_m = 0.05$ .....	58
Tabel 6.4 Uji Coba $N_{pop} = 200$ , $P_m = 0.05$ .....	58
Tabel 6.5 Uji Coba $N_{pop} = 300$ , $P_m = 0.05$ .....	59
Tabel 6.6 Uji Coba $N_{pop}=100$ , $P_c=0.2-0.9$ , $P_m=0.1$ .....	59
Tabel 6.7 Uji Coba $N_{pop} = 200$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	60
Tabel 6.8 Uji Coba $N_{pop} = 300$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	60
Tabel 6.9 Uji Coba $N_{pop} = 100$ , $P_c = 0.2-0.9$ , $P_m = 0.05$ .....	61
Tabel 6.10 Uji Coba $N_{pop} = 200$ , $P_c = 0.2-0.9$ , $P_m = 0.05$ .....	62
Tabel 6.11 Uji Coba $N_{pop} = 300$ , $P_c = 0.2-0.9$ , $P_m = 0.05$ .....	62
Tabel 6.12 Uji Coba $N_{pop} = 100$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	63
Tabel 6.13 Uji Coba $N_{pop} = 200$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	63
Tabel 6.14 Uji Coba $N_{pop} = 300$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	64
Tabel 6.15 Uji Coba $N_{pop} = 100$ , $P_m = 0.05$ .....	65
Tabel 6.16 Uji Coba $N_{pop} = 200$ , $P_m = 0.05$ .....	66
Tabel 6.17 Uji Coba $N_{pop} = 300$ , $P_m = 0.05$ .....	66
Tabel 6.18 Uji Coba $N_{pop}=100$ , $P_c=0.2-0.9$ , $P_m=0.1$ .....	67
Tabel 6.19 Uji Coba $N_{pop} = 200$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	68
Tabel 6.20 Uji Coba $N_{pop} = 300$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	69
Tabel 6.21 Uji Coba $N_{pop} = 100$ , $P_c = 0.2-0.9$ , $P_m = 0.05$ .....	69
Tabel 6.22 Uji Coba $N_{pop} = 200$ , $P_c = 0.2-0.9$ , $P_m = 0.05$ .....	70
Tabel 6.23 Uji Coba $N_{pop} = 300$ , $P_c = 0.2-0.9$ , $P_m = 0.05$ .....	71
Tabel 6.24 Uji Coba $N_{pop} = 100$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	72
Tabel 6.25 Uji Coba $N_{pop} = 200$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	72
Tabel 6.26 Uji Coba $N_{pop} = 300$ , $P_c = 0.2-0.9$ , $P_m = 0.1$ .....	73
Tabel 6.27 Hasil Terbaik Dari Masing-Masing Uji Coba Algoritma Genetika .....	76
Tabel 6.28 Total Jarak Terbaik Algoritma GA-TS .....	78
Tabel 6.29 Perbandingan Kedua Algoritma .....	78

Tabel 6.30 Hasil Terbaik Dari Masing-Masing Uji Coba Algoritma.....	80
Tabel 6.31 Total Jarak Terbaik Algoritma GA-TS.....	81
Tabel 6.32 Perbandingan Kedua Algoritma .....	82
Tabel 6.33 Validasi Rute Kendaraan Pertama Hari ke-1 Manual .....	84
Tabel 6.34 Validasi Rute Kendaraan Pertama Hari ke-2 Manual .....	84
Tabel 6.35 Validasi Rute Kendaraan Pertama Hari ke-3 Manual .....	85
Tabel 6.36 Validasi Rute Kendaraan Pertama Hari ke-4 Manual .....	86
Tabel 6.37 Validasi Rute Kendaraan Pertama Hari ke-5 Manual .....	87
Tabel 6.38 Validasi Rute Kendaraan Kedua Hari ke- 1 Manual	87
Tabel 6.39 Validasi Rute Kendaraan Kedua Hari ke-2 Manual	.88
Tabel 6.40 Validasi Rute Kendaraan Kedua Hari ke-3 Manual	.89
Tabel 6.41 Validasi Rute Kendaraan Kedua Hari ke-4 Manual	.89
Tabel 6.42 Validasi Rute Kendaraan Kedua Hari ke-5 Manual	.90
Tabel 6.43 Rute Awal Kendaraan Pertama .....	91
Tabel 6.44 Rute Awal Kendaraan Kedua .....	92
Tabel 9.1 Data Jarak (1) .....	<b>Error! Bookmark not defined.</b>
Tabel 9.2 Data Jarak (2) .....	<b>Error! Bookmark not defined.</b>
Tabel 9.3 Dataset Solomon.....	<b>Error! Bookmark not defined.</b>

## DAFTAR GAMBAR

Gambar 2.1.4 Diagram Alir Kombinasi Algoritma Genetika ....	22
Gambar 6.1 Rute Kendaraan Output Algoritma 1 .....	84
Gambar 6.2 Rute Kendaraan Output Algoritma 2 .....	85
Gambar 6.3 Rute Kendaraan Output Algoritma 3 .....	86
Gambar 6.4 Rute Kendaraan Output Algoritma 4 .....	86
Gambar 6.5 Rute Kendaraan Output Algoritma 5 .....	87
Gambar 6.6 Rute Kendaraan Output Algoritma 6 .....	88
Gambar 6.7 Rute Kendaraan Output Algoritma 7 .....	88
Gambar 6.8 Rute Kendaraan Output Algoritma 8 .....	89
Gambar 6.9 Rute Kendaraan Output Algoritma 9 .....	90
Gambar 6.10 Rute Kendaraan Output Algoritma 10 .....	90
Gambar 6.11 Rute Awal Depot Banyu Belik .....	93
Gambar 6.12 Rute Hari Pertama .....	94
Gambar 6.13 Rute Hari Kedua .....	95
Gambar 6.14 Rute Hari Ketiga .....	96
Gambar 6.15 Rute Hari Ke-Empat .....	97
Gambar 6.16 Rute Hari Ke-Lima .....	98
Gambar 6.17 Rute Hasil Optimasi .....	99
Gambar 6.18 Grafik Perbandingan Jarak .....	100
Gambar 9.1 Data Pengiriman dan Pengambilan Barang .....	<b>Error!</b>
<b>Bookmark not defined.</b>	
Gambar 9.2 Hasil Running (1) ...	<b>Error! Bookmark not defined.</b>
Gambar 9.3 Hasil Running (2) ...	<b>Error! Bookmark not defined.</b>
Gambar 9.4 Hasil Running (3) ...	<b>Error! Bookmark not defined.</b>
Gambar 9.5 Hasil Running (4) ...	<b>Error! Bookmark not defined.</b>

*Halaman ini sengaja dikosongkan*



# **BAB I**

## **PENDAHULUAN**

Pada bab ini akan dijelaskan mengenai latar belakang, rumusan masalah, batasan pengerjaan, tujuan, dan manfaat dari pengerjaan tugas akhir.

### **1.1. Latar Belakang Masalah**

Distribusi identik dengan memindahkan suatu barang dari pemasok hingga ke pelanggan akhir. Untuk dapat menyebarkan produk ke pelanggan akhir maka perusahaan diwajibkan melakukan aktifitas distribusi. Proses distribusi berpengaruh penting terhadap kepuasan konsumen, dimana konsumen akan meminta produk dikirimkan tepat waktu. Oleh karena itu proses distribusi juga menjadi proses yang penting bagi perusahaan. Proses distribusi dilakukan dalam berbagai jenis kendaraan untuk mengirimkan produk ke pelanggan melalui rute perjalanan yang telah ditentukan sebelumnya oleh perusahaan. Agar proses distribusi berjalan lancar maka dibutuhkan suatu perencanaan untuk dapat menentukan jadwal distribusi dan rute kendaraan yang sesuai.

Menentukan jadwal distribusi dan rute kendaraan bukan hal yang mudah, apabila terjadi penentuan rute kendaraan yang kurang tepat dapat mengakibatkan keterlambatan. Akhirnya muncul waktu tunggu dalam proses distribusi, keterlambatan di satu konsumen akan mempengaruhi waktu layanan pada konsumen yang selanjutnya jika dalam satu rute pengiriman yang dilalui kendaraan pengangkut terdapat lebih dari satu konsumen. Jika terjadi keterlambatan maka konsumen tidak puas dan merasa dirugikan [1].

Tidak hanya penentuan rute yang tepat agar dapat mengirimkan produk tepat waktu, permasalahan lain yang mungkin terjadi adalah ketersediaan kendaraan untuk pengiriman produk. Permasalahan ini tentu berdampak pada perusahaan. Dampak dari masalah tersebut dapat dirasakan tidak hanya oleh

perusahaan tetapi pelanggan juga ikut merasakan dampak dari masalah yang terjadi. Dampak yang terjadi dapat berpengaruh pada waktu pengiriman dan biaya. Waktu pengiriman menjadi tidak efektif karena banyak waktu yang dibuang karena adanya keterlambatan dan waktu tunggu baik di jalan maupun saat menunggu antrian di konsumen. Selain itu, dampak juga berpengaruh pada biaya karena kerugian-kerugian yang dapat dialami perusahaan. Penggunaan sumber daya yang tidak efektif dapat mengakibatkan peningkatan biaya untuk proses distribusi. Dari permasalahan tersebut, dibutuhkan suatu perhitungan yang sistematis untuk melakukan penentuan rute distribusi produk pada perusahaan [1].

Depot Air Minum Isi Ulang (Damiu) Banyu Belik yang berlokasi di Kabupaten Banyumas merupakan depot air minum yang sedang berkembang, untuk proses bisnisnya damiu tersebut juga mengirimkan produk air minum kemasan galonnya dengan prinsip pengiriman galon berisi air minum bersegel sekaligus pengambilan galon kosong dari pelanggan. Masalah yang dihadapi oleh damiu Banyu Belik sama seperti permasalahan yang telah dibahas di atas. Dari permasalahan tersebut maka damiu Banyu Belik membutuhkan optimasi rute pengiriman dan pengambilan produk untuk meminimalisir biaya proses bisnisnya.

Saat ini, Algoritma Genetika merupakan salah satu algoritma yang dicoba untuk mencapai solusi optimal TSP, walaupun sampai saat ini masih belum merupakan optimum global, algoritma ini sudah menghasilkan solusi yang sangat mendekati solusi optimum global [2].

Paper terkait dengan penelitian ini yang telah dibahas oleh peneliti sebelumnya tersaji pada tabel 1.1.

**Tabel 1.1 Jurnal Pendukung Penelitian**

<b>Judul Penelitian</b>	<b>Metode Penelitian</b>	<b>Tujuan Penelitian</b>	<b>Penulis</b>
<i>A Genetic Algorithm Based Approach to Vehicle Routing Problem with Simultaneous Pick-up and Deliveries [1]</i>	<i>Genetic Algorithm</i>	Membuktikan bahwa GA dapat memberikan solusi yang efisien untuk masalah penentuan rute pengiriman dan pengambilan barang.	(Tasan & Gen, 2012)
<i>A Hybrid GA-TS Algorithm for Open Vehicle Routing Optimization of Coal Mines Material [2]</i>	<i>Hybrid GA-TS Algorithm</i>	Membuktikan bahwa penggabungan metode GA-TS memberikan solusi yang lebih baik dibandingkan dengan menggunakan metode GA atau TS saja.	(Yu, Ding, & Zhu, 2011)

Oleh karena itu, pada penelitian ini dilakukan penyelesaian masalah menggunakan gabungan dari Algoritma Genetika dan Algoritma Pencarian Tabu, karena berdasarkan penelitian Yu, S., Ding, C., & Zhu, K. [3] menggunakan metode kombinasi Algoritma Genetika dan Pencarian Tabu menghasilkan solusi

terbaik dibandingkan dengan metode Algoritma Genetika atau Pencarian Tabu saja.

Menurut Fred Glover [5], ciri-ciri permasalahan yang dapat dikerjakan dengan menggunakan algoritma genetika adalah mempunyai multi kriteria dan multi objektif, mempunyai kemungkinan solusi yang jumlahnya banyak dan membutuhkan solusi yang dapat diperoleh dengan cepat. Namun algoritma genetika juga memiliki kelemahan, yaitu kurang mampu dalam pengidentifikasian solusi lokal sehingga proses pencarian mengalami pengulangan pada daerah solusi yang sama, hal ini menyebabkan solusi yang dihasilkan kurang variatif. Oleh sebab itu dibutuhkan kombinasi algoritma genetika dengan algoritma lainnya untuk menutupi kelemahan tersebut. Salah satunya adalah dengan kombinasi algoritma genetika dan pencarian tabu. Pencarian tabu menggunakan tabu list untuk menolak solusi-solusi yang memenuhi atribut tertentu guna mencegah proses pencarian mengalami cycling pada daerah solusi yang sama dan menuntun proses pencarian menelusuri daerah solusi yang belum dikunjungi.

Diharapkan dengan digunakannya kombinasi algoritma genetika dan pencarian tabu akan diperoleh optimasi penentuan rute kendaraan pada Damiu Banyu Belik.

## **1.2. Rumusan Masalah**

Permasalahan yang dititikberatkan pada tugas akhir ini adalah :

1. Bagaimana rute optimal untuk proses pengiriman Depot Air Minum Isi Ulang pada pelanggan aktif berdasarkan kombinasi algoritma genetika dan pencarian tabu?
2. Apakah penentuan rute menggunakan metode kombinasi algoritma genetika dan algoritma pencarian tabu menghasilkan rute yang lebih optimal daripada rute yang telah ditentukan Depot Air Minum Isi Ulang Banyu Belik?



### **1.3. Batasan Pengerjaan Tugas Akhir**

Batasan dalam pengerjaan tugas akhir ini adalah:

1. Sumber data yang digunakan adalah data dari Depot Air Minum Isi Ulang Banyu Belik Purwokerto, Kabupaten Banyumas, Jawa Tengah per tahun Februari 2015. Data yang digunakan meliputi data permintaan pelanggan aktif, alamat pelanggan aktif, armada mobil yang digunakan dan muatan armada.
2. Metode yang digunakan yaitu metode kombinasi algoritma genetika dan algoritma pencarian tabu.
3. Studi kasus yang diimplementasikan hanya menggunakan kendaraan mobil.

### **1.4. Tujuan Penelitian**

Tujuan pembuatan tugas akhir ini adalah untuk memperoleh rute distribusi Depot Air Minum Isi Ulang Banyu Belik kepada pelanggan aktif yang optimal.

### **1.5. Manfaat Penelitian**

Manfaat Manfaat yang akan diperoleh dengan adanya laporan ini antara lain:

- a. Memberikan solusi permasalahan rute kendaraan dengan mengoptimalkan rute pengiriman produk Depot Air Minum Isi Ulang Banyu Belik sehingga dapat meminimumkan biaya distribusi.
- b. Menambah wawasan peneliti mengenai kinerja dan penerapan algoritma genetika dikombinasikan dengan pencarian tabu terutama pada kasus penentuan rute optimal pada distribusi Depot Air Minum Isi Ulang Banyu Belik, Purwokerto.

*Halaman ini sengaja dikosongkan*

## **BAB II**

### **DASAR TEORI**

Pada bab ini dijelaskan mengenai teori-teori terkait yang bersumber dari buku, jurnal, ataupun artikel yang berfungsi sebagai dasar dalam melakukan pengerjaan tugas akhir agar dapat memahami konsep atau teori penyelesaian permasalahan yang ada.

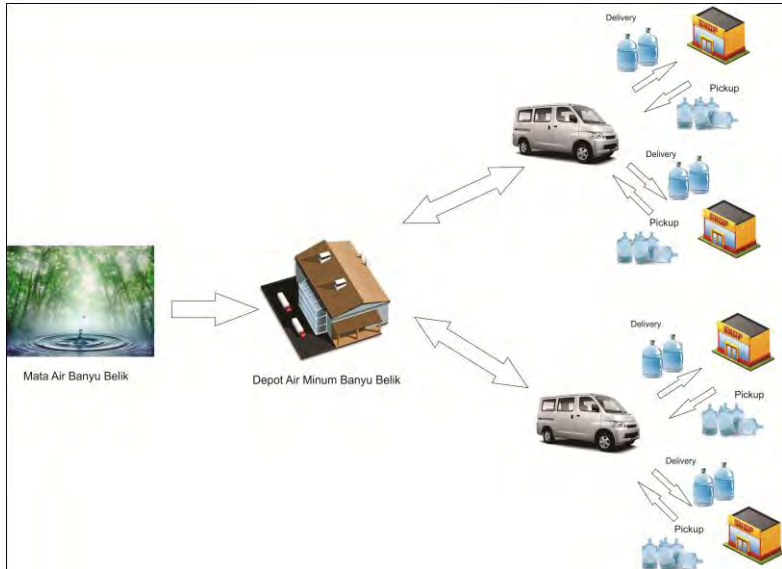
#### **2.1. Depot Air Minum Isi Ulang Banyu Belik**

Penelitian Depot Air Minum Isi Ulang (Damiu) Banyu Belik merupakan depot air minum yang berdiri sejak tahun 2010 yang terletak di Desa Karangnangka RT 01 RW 04, Kecamatan Kedungbanteng, Kabupaten Banyumas. Depot air minum ini milik dari Bapak Wasis Wardhana yang merupakan warisan dari keluarganya.

Asal usul dari nama “Banyu Belik” adalah berdasarkan dari nama mata air yang didapatkan dan digunakan dalam Depot Air Minum Banyu Belik ini. Untuk saat ini, Depot Air Banyu Belik menjual kurang lebih 4000 galon setiap bulannya. Target yang ingin dicapai dari Depot Air Minum Banyu Belik adalah mendapatkan status AMDK. Oleh karena itu, untuk merubah status, target yang harus dicapai oleh Banyu Belik adalah 10.000 galon penjualan per bulan.

Dalam distribusinya, Banyu Belik sudah mendistribusikan galonnya ke berbagai kota dan kabupaten di Jawa Tengah, antara lain adalah Kabupaten Purbalingga tepatnya di Kecamatan Pengadega dan Desa Banjarsari, Kota Cilacap, Kabupaten Banyumas, tepatnya di Kecamatan Sumpiuh, Cilongok, Banyumas dan masih banyak lainnya.

Untuk pendistribusiannya, Banyu Belik menggunakan 2 mobil Grandmax dengan maksimum angkutan 90 galon setiap mobil dan 2 motor Tossa dengan maksimum angkutan 16 galon setiap motornya. Untuk proses bisnisnya, bisa dilihat dalam gambar 2.1.



**Gambar 2.1 Proses Bisnis Damium Banyu Balik**

## **2.2. Vehicle Routing Problem (VRP)**

*Vehicle Routing Problem* (VRP) merupakan permasalahan yang di dalamnya ada sejumlah rute untuk sejumlah armada kendaraan yang berada pada satu atau lebih depot dan harus ditentukan jumlahnya guna melayani konsumen-konsumen yang tersebar. VRP adalah permasalahan optimasi kombinatorial kompleks yang merupakan gabungan dari dua model permasalahan yang terkenal yaitu Travelling Salesman Problem dan Bin Packing Problem. Model matematis dasar VRP memiliki fungsi tujuan untuk meminimumkan total jarak tempuh seluruh armada distribusi. [5]

VRP adalah problem yang masuk kategori NP-Hard Problem, yang berarti usaha komputasi yang digunakan akan semakin sulit dan banyak seiring dengan meningkatnya ruang lingkup masalah, dalam hal ini adanya peningkatan jumlah titik yang akan dilayani oleh kendaraan. [6].

### 2.3. *Vehicle Routing Problem with Pick-up and Delivering (VRPPD)*

*Vehicle Routing Problem (VRP) with Pick-up and Delivering* adalah sebuah VRP dimana kemungkinan pelanggan mengembalikan barang diperhitungkan. Maka di VRP-PD dibutuhkan perhitungan muatan barang untuk pengiriman dan pengambilan barang pada pelanggan. Parameter yang diperhitungkan adalah kapasitas kendaraan, jarak pengiriman atau banyaknya kendaraan yang dibutuhkan.

Tujuan dari memperhitungkan biaya distribusi adalah untuk meminimalkan armada kendaraan dan jumlah waktu perjalanan, dengan batasan bahwa kendaraan harus memiliki kapasitas yang cukup untuk mengangkut barang yang akan dikirim dan mengambil barang pada pelanggan untuk mengembalikan barang ke titik awal. Solusi dikatakan layak jika jumlah total ditugaskan untuk setiap rute tidak melebihi kapasitas kendaraan dan kendaraan memiliki kapasitas yang cukup untuk mengambil barang dari pelanggan. [7]

Dethloff menekankan pentingnya VRP-PD untuk kegiatan logistik dan ia menyelidiki hubungan antara VRP-PD dan jenis masalah routing lainnya. Ia mengusulkan formulasi matematika untuk masalah ini dan mengembangkan heuristik berbasis penyisipan, yang menggunakan beberapa kriteria penyisipan. [8]

Pada bagian berikut, VRP-PD diformulasikan sebagai model *mixed integer linier programming* berdasarkan model yang diberikan oleh Dethloff [8].

- Notasi:
  - J: kumpulan dari titik pelanggan
  - $J_0$ : kumpulan dari seluruh titik termasuk depot
  - V: kumpulan dari kendaraan

- Parameter:

cap: kapasitas/muatan kendaraan

$C_{ij}$ : jarak antara titik  $i$  dan  $j$   $i, j \in J_0, i \neq j$

$d_j$ : pengiriman jumlah permintaan berdasarkan titik pelanggan

$p_j$ : pengambilan jumlah permintaan dari titik pelanggan

$n$ : nomor titik

$M$ : angka yang cukup besar (contoh: nilai maksimum dari total pengiriman dan pengambilan barang atau jumlah jarak yang ditempuh oleh kendaraan pada setiap titik yang ada pada baris (1))

$$M = \max\{\sum_{j \in J} d_j + p_j\}, \sum_{i \in J_0} \sum_{j \in J_0, i \neq j} C_{ij} \quad (1)$$

- Variabel keputusan:

$L_v$ : Muatan dari kendaraan  $v$  ketika meninggalkan depot

$L_j$ : Muatan dari kendaraan setelah melayani pelanggan pada titik  $j$ ,  $j \in J$

$S_j$ : Interpretasi sebagai posisi dari titik  $j \in J$  di dalam rute

$X_{ijv}$ : Variabel *binary decision* yang diindikasikan ketika kendaraan  $v$  berjalan dari titik  $i$  ke  $j$

Fungsi tujuan dari VRP-PD yang ada pada baris (2) meminimasi total jarak perjalanan yang dilalui oleh kendaraan yang melayani pelanggan pada setiap titik.

$$\text{Minimize } \sum_{i \in J_0} \sum_{j \in J_0} \sum_{v \in V} C_{ij} X_{ijv} \quad (2)$$

**Tabel 2.1 Fungsi Batasan Model VRP-PD**

<b>Fungsi Batasan</b>	<b>Penjelasan</b>
$\sum_{i \in J_0} \sum_{v \in V} X_{ijh}, j \in J$	Setiap node hanya dilayani oleh satu armada
$\sum_{i \in J_0} X_{ikv} = \sum_{j \in J_0} X_{kjv},$ $k \in J, v \in V$	Tiap kendaraan akan masuk dan keluar dari kota yang sama.
$L'v = \sum_{i \in J_0} \sum_{j \in J} X_{kijv},$ $k \in J, v \in V$	Jumlah beban maksimal untuk masing-masing kendaraan saat meninggalkan depot.
$L_j \geq L'v - dj + pj - M(1 - X_{0jv}), j \in J, v \in V$ $L_j \geq L_i - dj + pj - M(1 - \sum_{v \in V} X_{ijv}), i \in J, j \in J, j \neq i$ $L'v \leq \text{cap}, v \in V$ $L_j \leq \text{cap}, j \in J$	Jumlah beban maksimal untuk masing-masing kendaraan setelah melayani node j.
$S_j \geq S_i + 1 - n(1 - \sum_{v \in V} X_{ijv}),$ $i \in J, j \in J, j \neq i$ $S_j \geq 0, j \in J$	Masing-masing armada hanya memiliki satu rute
$X_{ijv} \in \{0,1\}, i \in J_0, v \in V$	Variabel keputusan adalah bilangan biner.

## 2.4. Algoritma Genetika

Terdapat Algoritma Genetika adalah teknik pencarian dan optimasi berdasarkan prinsip proses genetika yang ada dalam makhluk hidup, yaitu perkembangan generasi dalam sebuah

populasi yang alami, secara lambat laun mengalami proses seleksi alam. Dengan meniru teori revolusi ini, Algoritma Genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan di dunia nyata. [9]

Pencipta Algoritma Genetika adalah John Holland. Algoritma Genetika menggunakan analogi secara langsung dari seleksi alam. Algoritma ini bekerja dengan sebuah populasi yang terdiri dari individu-individu, dimana masing-masing individu merepresentasikan sebuah solusi yang mungkin. Dalam Algoritma Genetika individu yang dimaksud dilambangkan dengan sebuah nilai fitness yang digunakan untuk mendapatkan nilai terbaik dari persoalan yang ada. Di dalam penerapan Algoritma Genetika akan melibatkan beberapa operator, yaitu:

- a) Operasi Evolusi yang melibatkan proses seleksi (selection) di dalamnya.
- b) Operasi Genetika yang melibatkan operator pindah silang (crossover) dan mutasi (mutation).

#### **2.4.1. Proses Algoritma Genetika**

Proses algoritma genetika dapat dilihat dalam diagram alir pada gambar 2.2. Algoritma genetika menggunakan pemilihan parent secara acak sehingga dapat menghasilkan keturunan dengan melalui operator genetika (contoh: crossover, mutation). Jumlah solusi potensial ditentukan oleh besarnya populasi yang ditentukan parameter acak sebelum sebelum solusi relovusi. Ketika generasi baru telah terbentuk, maka solusi tersebut akan melalui evaluasi dan terbentuk populasi baru untuk beberapa iterasi [10].

Prosedur umum teknik pencarian menggunakan algoritma genetika dijelaskan dalam tahapan berikut ini [11]:

- a. Membentuk populasi awal dengan membuat sejumlah kromosom yang mewakili solusi yang mungkin
- b. Menghitung nilai fitness



- c. Melakukan seleksi untuk memilih individu
- d. Melakukan operasi rekombinasi (crossover)
- e. Melakukan operasi mutasi
- f. Menghasilkan populasi dengan generasi baru, kembali ke tahap b.
- g. Ulangi proses sampai ditemukan individu dengan nilai fitness terbaik.

Dari tahapan di atas dapat terdapat beberapa istilah yang digunakan pada algoritma genetika. Berikut ini penjelasan dari operator-operator pada algoritma genetika:

#### a. Kromosom

Dalam algoritma genetika, tahapan awal yang perlu dilakukan adalah membentuk kromosom yang merupakan bagian dari populasi yang akan dibangkitkan. Setiap kromosom terbentuk dari gen-gen yang mewakili solusi yang mungkin. Dalam hal ini, kromosom merupakan variabel masukan dalam algoritma genetika [9]. Pada kasus penentuan rute kendaraan, kromosom harus berisi seluruh informasi dari suatu rute. Gen pada kromosom terdiri dari nomor kendaraan dan nomor urutan pelanggan dimana nilai yang mungkin dari gen adalah integer mewakili node permintaan atau node pelanggan. Nomor kendaraan mewakili kendaraan mana yang akan mengunjungi suatu node permintaan dan nomor urutan mewakili urutan kunjungan. Asumsi bahwa terdapat  $n$  node permintaan dengan set pelanggan  $N = \{1, 2, \dots, n\}$  dan set kendaraan  $K = \{1, 2, \dots, k\}$  dengan sejumlah kendaraan, kemudian panjang kromosom adalah  $n$ , dan posisi dalam kromosom adalah nomor dari node permintaan [12].

Sebagai contoh, terdapat delapan pelanggan untuk dikirim produk dengan permintaan yang berbeda-beda tiap pelanggan. Pertama-tama, hasilkan urutan integer antara 1 dan 8 seperti  $s = \{2, 5, 7, 3, 6, 4, 8, 1\}$  lalu dekode kromosom berdasarkan permintaan pelanggan dan kapasitas muatan tiap kendaraan.

Untuk lebih spesifik, urutan dimulai dari bagain paling kiri, kemudian menambah node satu demi satu sampai mencapai batas dari kapasitas muatan kendaraan. Penentuan rute dapat dihasilkan berdasarkan urutan kronologis dari penambahan node, dan rute selanjutnya bisa didapatkan jika proses dilanjutkan dari node yang tidak digabungkan. Proses diulang kembali sampai semua node dimasukkan sehingga didapatkan urutan  $\{0, 2, 5, 7, 0, 3, 6, 4, 0, 8, 1\}$ . Kromosom didekodekan dalam tiga rute antara lain [3]:

Kendaraan 1: depot  $\rightarrow 2 \rightarrow 5 \rightarrow 7$ ;

Kendaraan 2: depot  $\rightarrow 3 \rightarrow 6 \rightarrow 4$ ;

Kendaraan 3: depot  $\rightarrow 8 \rightarrow 1$ .

#### b. Populasi

Algoritma genetika dimulai dari sekumpulan kromosom yang diketahui sebagai populasi. Inisialisasi populasi pada umumnya memiliki dua mekanisme. Salah satunya adalah generasi acak. Pada kasus ini, penentuan populasi mengadopsi generasi acak, akan tetapi tidak diperbolehkan ada solusi yang identik pada populasi yang dibentuk [13].

#### c. Fungsi fitness

Solusi yang dihasilkan perlu dilakukan pengukuran agar solusi yang dihasilkan bisa sesuai dengan tujuan yang diinginkan. Fungsi fitness merupakan komponen algoritma genetika dalam mengukur tingkat kebaikan atau kesesuaian pada solusi yang diberikan. Dalam kasus ini ada fungsi fitness atau nilai fitness yang digunakan adalah untuk meminimumkan jarak dan biaya.

#### d. Seleksi

Tahapan setelah menentukan nilai fitness adalah proses seleksi. Operator ini memilih kromosom yang ada dalam populasi untuk proses reproduksi. Operator ini mereplikasi solusi yang paling sukses yang ditemukan dalam populasi

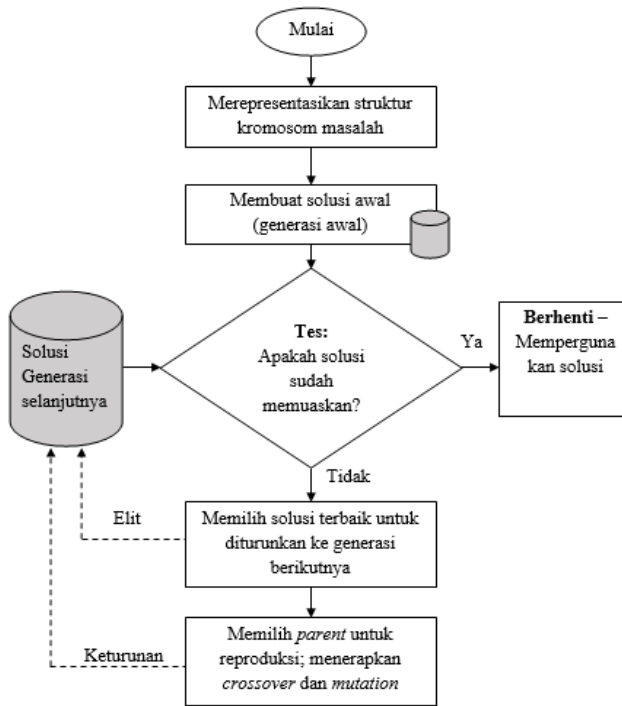
pada tingkat yang proporsional untuk kualitas relatif yang diinginkan. Sebaik kromosom yang sesuai, semakin banyak waktu untuk dipilih untuk reproduksi [14].

e. Crossover (rekombinasi)

Crossover atau kawin silang adalah proses probabilistik yang menukar informasi antara dua induk untuk menghasilkan dua kromosom anak [15]. Operator ini secara acak memilih suatu lokus dan menukar subsekuen sebelum dan sesudah bahwa lokus antara dua kromosom untuk menciptakan dua keturunan. Sebagai contoh, string 10000100 dan 11111111 dapat di pindahkan setelah lokus ketiga dari keduanya untuk membuat dua keturunan 10011111 dan 11100100. Operator crossover secara kasar menirukan rekombinasi biologis antara dua organisme kromosom tunggal [14].

f. Mutasi

Operator ini secara acak membalik beberapa bit dalam suatu kromosom. Sebagai contoh, string 00000100 mungkin diganti pada posisi keduanya untuk menghasilkan 01000100. Mutasi dapat terjadi pada setiap posisi bit dalam suatu string dengan beberapa kemungkinan yang biasanya sangat kecil [14].



**Gambar 2.2 Diagram Alir Algoritma Genetika (Sumber: Turban, 2010)**

## 2.5. Algoritma Pencarian Tabu

Pencarian Tabu diusulkan pertama kali dan dikembangkan oleh Fred Glover pada tahun 1986 terinspirasi dari prinsip kecerdasan buatan [16]. Pencarian tabu adalah sebuah metode optimasi yang berbasis pada local search. Proses pencarian bergerak dari satu solusi ke solusi berikutnya, dengan cara memilih solusi terbaik neighbourhood solusi sekarang (current) yang tidak tergolong solusi.

Pencarian Tabu diusulkan pertama kali dan dikembangkan oleh Fred Glover pada tahun 1986 terinspirasi dari prinsip kecerdasan buatan [16]. Pencarian tabu adalah sebuah metode

optimasi yang berbasis pada local search. Proses pencarian bergerak dari satu solusi ke solusi berikutnya, dengan cara memilih solusi terbaik *neighbourhood* solusi sekarang (current) yang tidak tergolong solusi.

Ide dasar dari algoritma pencarian tabu adalah mencegah proses pencarian dari local search agar tidak melakukan pencarian ulang pada ruang solusi yang sudah pernah ditelusuri, dengan memanfaatkan suatu struktur memori yang mencatat sebagian jejak proses pencarian yang telah dilakukan.

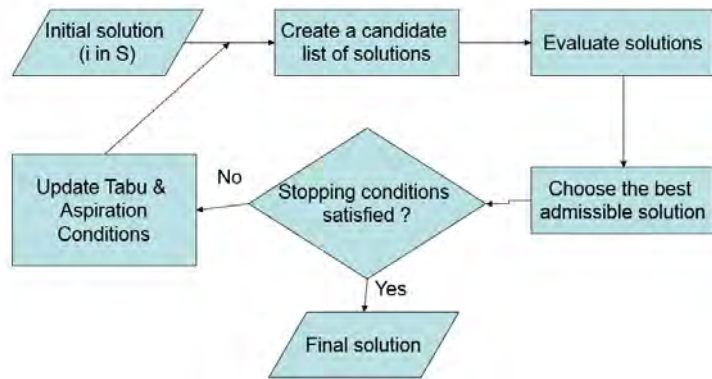
Struktur memori fundamental dalam pencarian tabu dinamakan tabu list. Tabu list menyimpan atribut dari sebagian move (transisi solusi) yang telah diterapkan pada iterasi-iterasi sebelumnya. Pencarian tabu menggunakan tabu list untuk menolak solusi-solusi yang memenuhi atribut tertentu guna mencegah proses pencarian mengalami cycling pada daerah solusi yang sama, dan menuntun proses pencarian menelusuri daerah solusi yang belum dikunjungi. Tanpa menggunakan strategi ini, local search yang sudah menemukan solusi optimum lokal dapat terjebak pada daerah solusi optimum lokal tersebut pada iterasi-iterasi berikutnya. [16]

List ini mengikuti aturan LIFO dan biasanya sangat pendek (panjangnya biasanya sebesar  $O(\sqrt{N})$ , dimana  $N$  adalah jumlah total dari operasi) [5]. Setiap saat ada langkah itu akan ditempatkan dalam tabu list. Perekaman solusi secara lengkap dalam sebuah forbidden list dan pengecekan apakah sebuah kandidat solusi tercatat dalam list tersebut merupakan cara yang mahal, baik dari sisi kebutuhan memori maupun kebutuhan waktu komputasi. Jadi, tabu list hanya menyimpan langkah transisi (move) yang merupakan lawan atau kebalikan dari langkah yang telah digunakan dalam iterasi sebelumnya untuk bergerak dari satu solusi ke solusi berikutnya. Dengan kata lain tabu list berisi langkah-langkah yang membalikan solusi yang baru ke solusi yang lama. Pada setiap iterasi, dipilih solusi baru yang merupakan solusi terbaik dalam *neighbourhood* dan tidak tergolong sebagai tabu. Kualitas solusi baru ini tidak harus lebih

baik dari kualitas solusi sekarang. Apabila solusi baru ini memiliki nilai fungsi objektif lebih baik dibandingkan solusi terbaik yang telah dicapai sebelumnya, maka solusi baru ini dicatat sebagai solusi terbaik yang baru. Sebagai tambahan dari tabu list, dikenal adanya kriteria aspirasi, yaitu suatu Penanganan khusus terhadap move yang dinilai dapat menghasilkan solusi yang baik akan tetapi move tersebut berstatus tabu. Dalam hal ini, jika move tersebut memenuhi criteria aspirasi yang telah ditetapkan sebelumnya, maka perpindahan tersebut dapat digunakan untuk membentuk solusi berikutnya (status tabunya dibatalkan). Gambar 2.3 merupakan diagram alir dari algoritma pencarian tabu. [5]

Dasar-dasar proses algoritma pencarian tabu [16]:

- 1) Pilih sebuah solusi  $I$  dalam  $S$ . Set  $i^* = I$  dan  $k = 0$
- 2) Set  $k=k+1$  dan menghasilkan sebuah subset  $V^*$  dari solusi dalam  $N(i,k)$  sehingga salah satu kondisi tabu dilanggar atau lebih dari satu kondisi aspirasi terpilih
- 3) Pilih  $j$  terbaik dalam  $V^*$  dan set  $i=j$
- 4) If  $f(i^*) > f(i)$  then set  $i^* = i$
- 5) Memperbarui tabu dan kondisi aspirasi
- 6) Jika sebuah solusi selesai maka berhenti, jika tidak kembali ke step 2.



**Gambar 2.3 Diagram Alir Pencarian Tabu (Sumber: Glover & Laguna, 1997)**

## **2.6. Kombinasi Algoritma Genetika dan Pencarian Tabu**

Algoritma genetika dan algoritma pencarian tabu secara biasa dipandang berdasarkan fondasi dan sudut pandang. Walaupun ada beberapa perbedaan yang signifikan antara kedua metode ini, kedua pendekatan ini berbagi elemen tertentu yang sering tidak dilihat dan tidak dipergunakan. Walaupun memiliki beberapa perbedaan, salah satunya adalah perbedaan dalam mekanisme yang digunakan, kedua algoritma ini memiliki elemen yang dapat menguatkan proses pencarian solusi. Dalam penelitiannya, Fred Glover dkk menyatakan bahwa dalam kerangka kerja algoritma pencarian tabu memiliki gelombang strategik (strategic oscillation). Gelombang strategik ini merupakan sebuah proses yang biasanya mengubah-ubah arah dari pencarian dan wilayah dikunjungi dengan mengontrol pergerakan yang diperbolehkan atau yang dievaluasi sekitar solusi saat ini. Penggunaan yang paling berguna dari elemen ini adalah untuk menciptakan proses pencarian yang luas yang dapat mengijinkan solusi infisibel atau yang tidak dapat dijalankan selama pencarian.

Secara umum algoritma genetika tidak mengijinkan operasi rekombinasi atau crossover yang akan menghasilkan solusi infisibel atau solusi yang tidak dapat dijalankan. Dalam kondisi khusus dimana solusi ini diperbolehkan, solusi ini biasanya diubah menjadi solusi yang fisibel sebelum mereka ditempatkan kembali ke dalam populasi. Gelombang strategik dalam algoritma pencarian tabu dapat memperbolehkan anak yang infisibel hidup dipopulasi tanpa harus diubah terlebih dahulu ke bentuk fisibel. Dengan memperkaya populasi dengan solusi infisibel, algoritma genetika dipasangkan dengan gelombang strategik memperoleh kekuatan untuk beroperasi dengan ragam yang meningkat yang mungkin mengembangkan kesempatan untuk menemukan solusi yang optimum [17]. Beberapa peneliti cukup banyak yang telah menggabungkan algoritma genetika dan algoritma pencarian tabu dalam menyelesaikan masalah penentuan rute pengiriman. Thamilselvan dan Balasubramanie menggabungkan kedua algoritma ini untuk menyelesaikan masalah penjadwalan kerja dan masalah perjalanan salesman dimana pada proses seleksi di algoritma genetika menggunakan proses pencarian tabu [18]. Yu, S., Ding, C. & Zhu, K menggabungkan algoritma genetika dan pencarian tabu pada permasalahan penentuan rute kendaraan dengan ide dasar untuk menggabungkan perhitungan paralel dan optimasi global yang dimiliki algoritma genetika dengan kemampuan pencarian local yang cepat yang dimiliki oleh pencarian tabu [3].

Berdasarkan beberapa contoh referensi di atas, beberapa masalah yang ada diangkat menjadi landasan untuk menyelesaikan masalah penentuan rute kendaraan pada Depot Air Minum Isi Ulang Banyu Belik. Referensi yang digunakan dalam tugas akhir ini adalah penelitian yang dilakukan oleh Thamilselvan dan Balasubramanie serta penelitian yang dilakukan oleh Yu, S., Ding, C. & Zhu, K sebagai landasan karena memiliki beberapa kesamaan dengan kasus yang diusung.



Pada penelitian yang dilakukan oleh Yu, S., Ding, C. & Zhu, K, secara sederhana kromosom dibentuk dari urutan (permutasi)  $s$  dari sejumlah  $n$  node konsumen. Karakter yang digunakan adalah set kode integer yang berisi angka alami dari urutan  $\{1, 2, 3, \dots, n\}$ . Pengaturan kode integer yang memasukkan node konsumen dan simbol kendaraan menyatakan rute dari tiap kendaraan lebih sesuai untuk permasalahan ini daripada kode biner [3].

Langkah awalnya adalah menghasilkan urutan integer kemudian menentukan kromosom berdasarkan permintaan konsumen dan kapasitas muatan kendaraan. Setelah kromosom ditentukan, proses selanjutnya disesuaikan dengan proses pada algoritma genetika sampai pada hasil atau solusi rute yang paling optimum ditemukan.

## **2.7. Mekanisme Kombinasi Algoritma Genetika dan Pencarian Tabu**

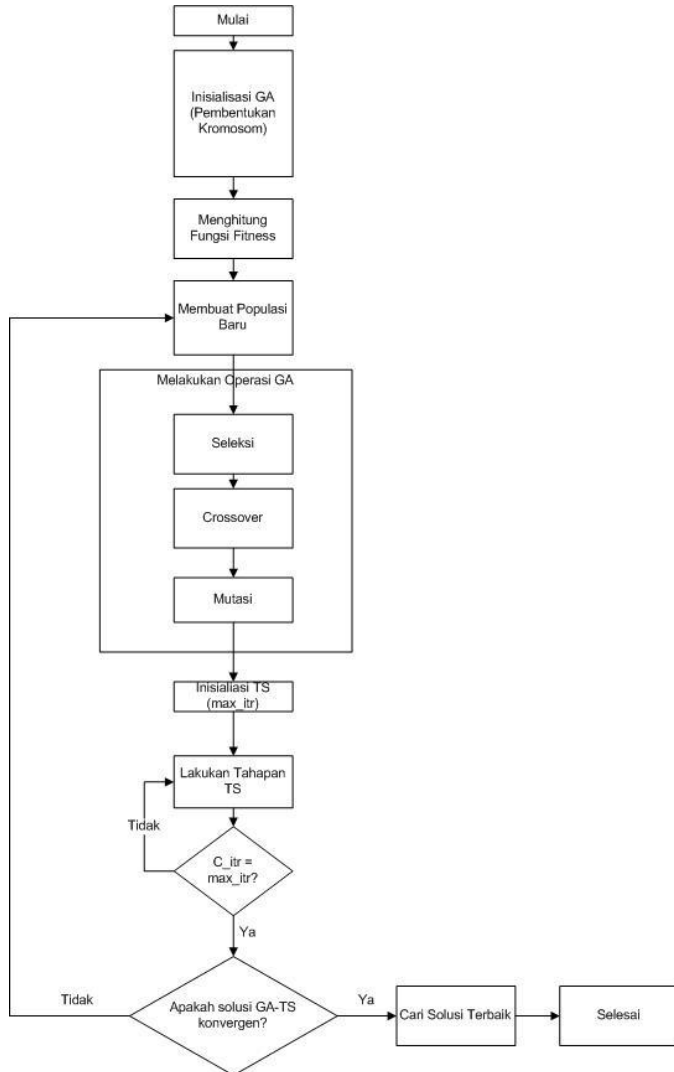
Seperti yang telah dikatakan oleh Yu, et. al bahwa ide dasar dari optimasi masalah penjadwalan dan penentuan rute kendaraan dengan menggunakan kombinasi algoritma genetika dan algoritma pencarian tabu adalah menggabungkan perhitungan paralel algoritma genetika dan optimasi global dari kemampuan pencarian tabu dan pencarian lokal cepat dari algoritma pencarian tabu. Dengan kata lain, algoritma menggunakan pengkodean angka alami, yang individu-individu dari populasi melakukan pencarian lokal menggunakan pencarian tabu dengan tingkat tertentu dari probabilitas sehingga tidak hanya menjamin optimasi secara global namun juga meningkatkan kecepatan dalam perhitungan [3].

Perbandingan antara algoritma genetika dan algoritma pencarian tabu telah menunjukkan jika keduanya memiliki kekurangan dan kelebihanannya sendiri. Algoritma Genetika menemukan solusi lebih cepat jika berada dekat pada solusi terbaik, namun algoritma genetika dapat jatuh pada lokal minima lebih mudah. Secara kontras berbeda dengan pencarian

tabu, yang dapat mencegah jatuh ke lokal minima namun kecepatan penemuannya yang lebih lambat daripada algoritma genetika [3]. Oleh karena itu algoritma gabungan dibuat.

Variasi gabungan antara algoritma genetika dan algoritma pencarian tabu yang digunakan pada tugas akhir ini adalah menggunakan algoritma pencarian tabu setelah algoritma genetika mengeluarkan nilai fitness. Apabila nilai fitness setelah menggunakan algoritma pencarian tabu lebih baik maka digunakan nilai fitness tersebut, apabila tidak kembali ke nilai fitness yang dihasilkan algoritma genetika.

Gambar 2.4 merupakan diagram alir dari kombinasi algoritma genetika dan pencarian tabu yang telah disesuaikan dengan studi kasus.



**Gambar 2.1.4 Diagram Alir Kombinasi Algoritma Genetika dan Pencarian Tabu (Sumber: K.L Katsifarakis)**

*Halaman ini sengaja dikosongkan*

## **BAB III**

### **METODOLOGI Pengerjaan Tugas Akhir**

Pada bagian ini menjelaskan mengenai tahapan yang dilakukan dalam pengerjaan tugas akhir. Alur pengerjaan yang dilakukan dalam tugas akhir ini digambarkan pada Gambar 3.1. Berikut ini merupakan detail tahapan dari metodologi yang dilakukan pada tugas akhir antara lain penyiapan data yang terdapat dua proses yaitu pengumpulan data dan praproses data, implementasi metode, verifikasi dan validasi, uji coba dan analisis, dokumentasi tugas akhir.

#### **3.1. Penyiapan Data**

Pada tahap penyiapan data terdapat dua proses, yaitu yang pertama proses pengumpulan data dan yang kedua praproses data. Penjelasan dari dua proses tersebut dijelaskan pada sub-bab selanjutnya.

##### **3.1.1. Pengumpulan Data**

Pada tahap ini dilakukan pengumpulan data dan informasi yang dibutuhkan dalam pelaksanaan tugas akhir yang berkenaan dengan proses distribusi pada Depot Air Minum Banyu Belik terkait dengan data pelanggan meliputi informasi-informasi terkait dengan pelanggan aktif, data kendaraan meliputi informasi-informasi terkait dengan kendaraan yang dimiliki oleh damiu, dan data banyaknya permintaan pelanggan yang lokasinya dikirimkan dengan kendaraan mobil.

Cara yang digunakan untuk mengumpulkan data dan informasi yaitu interview dengan Bapak Wasis Wardhana selaku pemilik dari damiu Banyu Belik. Hasil yang didapatkan dari tahap ini adalah mendapatkan data pelanggan aktif, data kendaraan

beserta muatannya, lokasi pengiriman yang dikirim dengan kendaraan mobil.

Untuk validasi sebuah algoritma maka diperlukan perhitungan ulang secara manual menggunakan jarak sebenarnya. Dalam tugas akhir ini menggunakan Microsoft Excel. Apabila menghasilkan nilai jarak yang sama, maka dikatakan valid

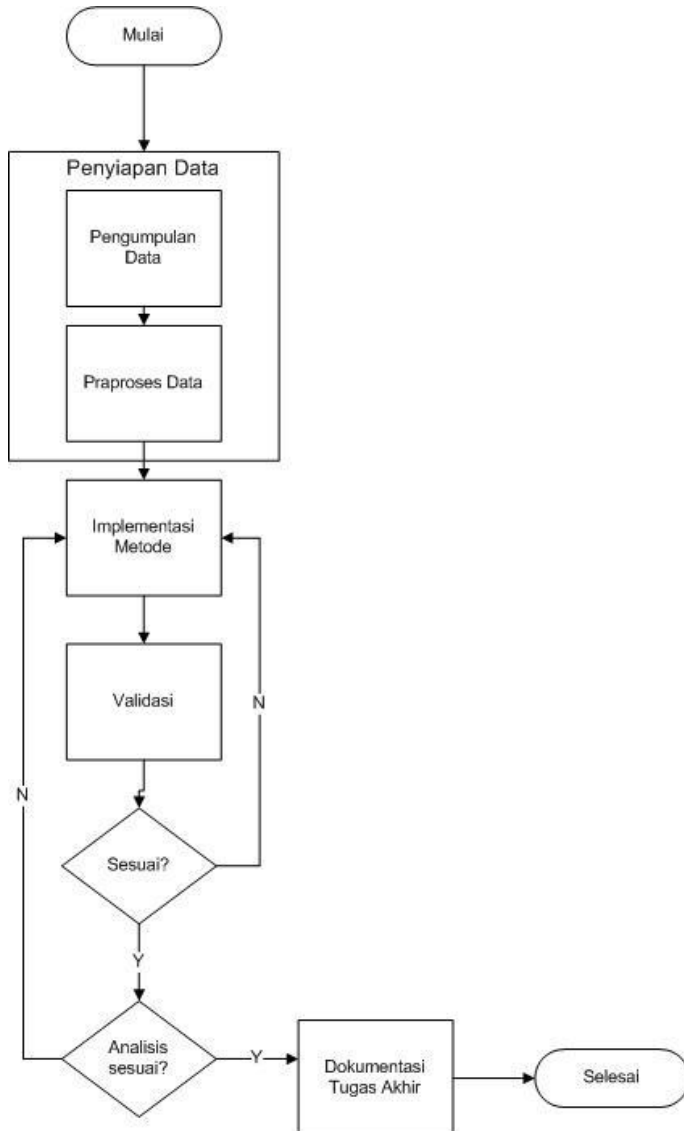
### **3.1.1. Praproses Data**

Pada tahap ini, data yang didapatkan dari hasil pengumpulan data dan informasi diproses terlebih dahulu, yaitu dengan cara pembersihan data agar tidak ada masalah pada kualitas data seperti modifikasi data yang asli, duplikasi data, terdapat data yang memiliki karakteristik yang berbeda, terdapat data yang kosong dan lain-lain. Hasil yang didapatkan pada tahap ini merupakan data yang siap digunakan pada tahap berikutnya.

### **3.2. Implementasi Metode**

Pada tahap ini, merancang algoritma genetika dan pencarian tabu dengan menggunakan fungsi tujuan dan batasan sesuai dengan Dethloff's model yang telah dijelaskan pada dasar teori 2.3.

Hasil yang didapatkan pada tahap ini adalah implementasi model matematika ke dalam kombinasi algoritma genetika dan pencarian tabu.



**Gambar 3.1 Metodologi Pengerjaan Tugas Akhir**

### **3.3. Validasi**

Pada tahap ini, algoritma yang telah dibuat pada tahap sebelumnya dilakukan validasi model pendekatan untuk membandingkan hasil dari algoritma dengan hasil di penelitian sebelumnya dengan menggunakan dataset yang sama. Dataset yang digunakan yaitu RC105 Solomon [3] lalu dibandingkan dengan hasil dari penelitian Jean Berger, Mohamed Barkaoui yang berjudul “*A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem*”. Untuk validasi hasil dilakukan perbandingan dengan total jarak dari perhitungan manual menggunakan Excel, sehingga dapat dibandingkan dengan total jarak yang dihasilkan algoritma. Apabila hasilnya sama, maka dapat dikatakan algoritma tersebut valid.

### **3.4. Analisis Hasil**

Analisis hasil berisi penjelasan tentang semua hasil output yang didapat dari validasi untuk mengetahui apakah algoritma yang dibuat telah sesuai dengan tujuan yang telah ditentukan. Kemudian dibandingkan dengan rute yang ditentukan perusahaan sebelumnya. Hasil yang didapatkan dari tahapan ini adalah rute optimal berdasarkan hasil analisis.

### **3.5. Dokumentasi Tugas Akhir**

Tahapan terakhir pada penelitian tugas akhir adalah dokumentasi dari seluruh tahapan pengerjaan tugas akhir serta pembahasan penelitian ke dalam buku tugas akhir. Hasil dari tahapan ini adalah terbentuknya buku tugas akhir.



## **BAB IV PERANCANGAN**

Pada bab ini akan dijelaskan bagaimana rancangan dari penelitian tugas akhir yang meliputi subyek dan obyek dari penelitian, pemilihan subyek dan obyek penelitian dan bagaimana penelitian akan dilakukan.

### **4.1. Deskripsi Data**

Data yang akan digunakan pada penelitian ini adalah data dari studi kasus serta data set dari Solomon yang akan digunakan untuk menguji kemampuan kombinasi algoritma genetika dan pencarian tabu. Atribut dari data tersebut adalah data jarak dalam satuan kilometer atau dalam bentuk matriks, permintaan pelanggan, dan kendaraan untuk mengantar dan mengambil barang.

Berikut ini adalah variable input dari data dari studi kasus beserta asumsinya yang akan digunakan pada saat menjalankan algoritma:

- a. Data permintaan galon studi kasus Banyu Belik yang dikirim dan diambil bernilai sama karena pada proses bisnis perusahaan merupakan kewajiban untuk memastikan galon yang terisi dan siap dikirim jumlahnya sama dengan galon kosong yang akan diambil oleh Damiu. Data permintaan tersebut terdapat pada M-file yang bernama `problem_nisa.m`. Nantinya pada kombinasi algoritma genetika dan pencarian tabu menggunakan file `problem_nisa.m` sebagai variable input.

**Tabel 4.1 Data permintaan pengiriman dan pengambilan galon**

No Depot	Alamat	Delivery	Pick Up
<b>1</b>	Kec. Cilongok	45	45
<b>2</b>	Ds. Pasir Kec Karang lewas	35	35
<b>3</b>	Ds. Kedungbanteng Kecamatan Kedung banteng	30	30
<b>4</b>	Kec. Rawalo	35	35
<b>5</b>	Kec. Jatilawang	35	35
<b>6</b>	Kec. Banyumas	45	45
<b>7</b>	Kec. Sumpiuh	50	50
<b>8</b>	Kec. Kalibagor	45	45
<b>9</b>	Kec. Sokaraja	15	15
<b>10</b>	Kec. Somagede	30	30
<b>11</b>	Perumahan Teluk Purwokerto	45	45
<b>12</b>	Tanjung, Purwokerto	25	25
<b>13</b>	Karang Jambu, Purwokerto	40	40
<b>14</b>	Grendeng, purwokerto	40	40
<b>15</b>	Banjar sari , Purbalingga	50	50
<b>16</b>	Pengadegan , Purbalingga	60	60

No Depot	Alamat	Delivery	Pick Up
<b>17</b>	Jl lombok - kel. gunung simping	35	35
<b>18</b>	Jl ternate - kel.gunung simping	45	45
<b>19</b>	Teritih lor , kec jeruk legi	30	30
<b>20</b>	Jl sri gunting - kel. sidanegara	30	30
Total galon yang diambil dan dikirim harus sama		<b>765</b>	<b>765</b>

- b. Data jarak antar node dalam bentuk matriks yang dilampirkan pada lampiran A dan disimpan dengan nama jarak\_node\_nisa.m sebagai M-file. Nantinya kombinasi algoritma GA dan TS akan memanggil file tersebut sebagai variable input.
- c. Armada yang digunakan untuk mengirim dan mengambil barang yaitu mobil Grand Max dengan kapasitas kendaraan yaitu 90 galon dan motor box Tossa dengan kapasitas 16 galon. Yang akan digunakan pada penelitian tugas akhir ini hanya kendaraan mobil. Dengan asumsi kendaraan mobil yang dimiliki adalah dua buah, apabila tidak memadai maka barang dikirimkan pada hari selanjutnya sesuai dengan permintaan perusahaan yang tidak menginginkan adanya pengadaan kendaraan. Penamaan variable input untuk kapasitas maksimal kendaraan dalam algoritma adalah kapasitas\_kendaraan.

## 4.2. Desain Permodelan Permasalahan

Sebelum merancang kombinasi algoritma genetika dan pencarian tabu dibutuhkan perhitungan fungsi tujuan dan perhitungan asumsi atau batasan yang dibutuhkan seperti jumlah armada dan kapasitas kendaraan. Fungsi batasan lain sesuai dengan permasalahan *Vehicle Routing Problem Pick Up and Delivery* yang telah dijelaskan pada sub-bab 2.2. Berikut kode yang dirancang:

- a. Sintaks function pada code 4.1 di bawah ini mendeklarasikan bahwa M-File yang digunakan bernama “Constraints\_GA” yang jika dijalankan membutuhkan variable input yaitu “problem\_nisa”, “jarak\_node”, “kapasitas\_kendaraan”, “F”, “CR”, “pop\_size”, “max\_iterasi”

```
function [totaljarak routes jumlah_hari] =
Constraints_GA(problem,jarak_node,kapasitas
kendaraan, route)
```

**Code 4.1 Sintaks Function**

- b. Berikut pendefinisian variable input Constraints\_GA seperti yang ditunjukkan code

```
[problem] = problem;
[n m]= size (problem);
a = meshgrid(1:n);
[jarak_node] = jarak_node;
deliveries = problem(:,2);
pickup = problem(:,3);
kapasitas=kapasitas_kendaraan;
```

**Code 4.2 Definisi Variabel Input**

- c. Perhitungan yang pertama adalah Evaluasi Kendala Kapasitas Kendaraan, ditunjukkan pada code 4.3. Dalam satu kromosom, saat kapasitas yang sedang diangkut melebihi kapasitas maksimal kendaraan, kromosom akan langsung dipotong menjadi 1 rute.

```

kota=route(cc);

rutenya{hh,ee}(ff)=route(hh,cc);

yang_dikirim(hh,ee)=yang_dikirim(hh,ee)+de
liveries(route(hh,cc));

yang_diambil(hh,ee)=yang_diambil(hh,ee)+pi
ckup(route(hh,cc));
                cc=cc+1;
                ff=ff+1;

current_load_next=yang_diambil(hh,ee)+deli
veries(route(hh,cc));
                end
                rutenya{hh,ee}(ff)=1;
                ee=ee+1;
                kendaraan_ke=kendaraan_ke+1;
        end
        hh=hh+1;
end
routes=rutenya;
a=routes;

```

**Code 4.3 Evaluasi Kendala**

- d. Code 4.4 menunjukkan perhitungan yang kedua yaitu perhitungan jumlah hari yang dibutuhkan. Pengiriman dilakukan hanya dilakukan 1x rute perjanan dalam sehari, yaitu dari depot ke depot lagi. Pada code 4.4 ini. Menambah hari terjadi apabila length dari rute yang

terbentuk adalah  $> 2$  node. Jika  $\leq 2$ , node yang ada dalam rute tersebut paling banyak ada 2 yaitu dari node 1 ke node 1. Artinya, di dalam rute tersebut tidak terjadi perjalanan. Misal, apabila terbentuk 5 rute dalam 1 kromosom, maka waktu yang dibutuhkan adalah 5 hari.

```

jumlah_hari = 2;
[brs klm]=size(a);
matrik_keb_hari=[];
for jumlah_pop=1:brs
    kebutuhan=0;
    for jumlah_string=1:klm
        if
length(rutenya{jumlah_pop,jumlah_string})
<=2
            tambah_hari=0;
        else
            tambah_hari=1;
        end

    kebutuhan=kebutuhan+tambah_hari/jumlah_ha
ri;
    kebutuhan(kebutuhan==5.5) = 6;
end

matrik_keb_hari=[matrik_keb_hari;kebutuha
n];
end
matrik_keb_hari=matrik_keb_hari;
[abc xyz]=size(routes);

```

#### Code 4.4 Perhitungan Hari

Karena jumlah kendaraan yang digunakan untuk mengirim dan mengambil adalah 2, maka waktu yang dibutuhkan menjadi  $5\text{hari}/2$ , yaitu 2,5 hari. Rumus perhitungan waktu atau hari yang dibutuhkan adalah sebagai berikut:

- e. Code 4.5 menunjukkan perhitungan yang ketiga adalah perhitungan jarak perjalanan. Rute perjalanan dihitung berdasarkan kromosom atau rute yang sudah terbentuk pada saat evaluasi kapasitas kendaraan. [3]

```

clear total_jaraknya
total_jarak_perjalanan=[];

for jumlah_pop=1:abc
    biaya_bbm_per_jarak_per_kend=[];
    totaljarakz2=[];
clear total_jaraknya
total_jarak_perjalanan=[];

for jumlah_pop=1:abc
    biaya_bbm_per_jarak_per_kend=[];
    totaljarakz2=[];
for jumlah_string=1:xyz
    if
length(rutenya{jumlah_pop,jumlah_string})
<=2
        Rute_Akhir=[];
    else

Rute_Akhir=rutenya{jumlah_pop,jumlah_string};

        end
        rute_dievaluasi=Rute_Akhir;
        idxktskr=1;

bw_del=sum(deliveries(rute_dievaluasi));
muatan_skr=bw_del;
ccc=jumlah_string;
    if length(rute_dievaluasi)>=3
        while
idxktskr<length(rute_dievaluasi)

```

**Code 4.5 Perhitungan Jarak Perjalanan**

### 4.3. Desain Algoritma

Pada sub-bab ini bertujuan untuk mengetahui alur kerja kombinasi algoritma genetika dan pencarian tabu. Berikut



merupakan desain kombinasi algoritma genetika dan pencarian tabu untuk menyelesaikan permasalahan penentuan rute kendaraan pada studi kasus pengiriman barang dalam bentuk galon ke pelanggan.

#### 4.3.1. Desain Algoritma Genetika

Di dalam algoritma genetika terdapat beberapa langkah untuk mendapatkan nilai optimum seperti yang dijelaskan pada subbab 2.4. Di dalam tugas akhir ini menggunakan *tool* optimasi MATLAB 8.3.0.

Berikut merupakan penjelasan dalam setiap proses yang dilakukan pada algoritma genetika [4].

- Inisialisasi.  
Di dalam langkah inisialisasi menentukan jumlah gen dalam satu kromosom, yaitu sebanyak kota. Menentukan ukuran populasi, probabilitas crossover, dan probabilitas mutasi.  
Pada penelitian ini ditentukan sebagai berikut berdasarkan penelitian sebelumnya [3]  
Ukuran Populasi = 100, 200, 300  
Probabilitas *crossover* = 0.2-0.9  
Probabilitas mutasi = 0.1 dan 0.05
- Membangkitkan populasi awal.  
Membangkitkan sejumlah rute sebanyak ukuran populasi secara random. [4]
- Melakukan perhitungan nilai *fitness*.  
Perhitungan nilai fitness dilakukan dengan menghitung jarak total dari rute yang dihasilkan. Pilih kromosom terbaik yang memiliki nilai fitness terkecil yang artinya jarak yang dilalui terpendek. [4]
- Elitisme.  
Melakukan penyalinan individu terbaik ke dalam populasi berikutnya. Dalam hal ini yaitu salah satu usaha mempertahankan individu-individu terbaik yang

telah diperoleh di suatu generasi ke dalam generasi berikutnya. Sehingga individu-individu terbaik ini akan tetap muncul di populasi berikutnya. [4]

- Seleksi dengan roda lotere.  
Seleksi dengan roda lotere atau yang sering disebut dengan *roulette wheel* untuk menentukan induk pada proses *crossover* dan menentukan kromosom yang akan mengalami mutasi. [4]
- *Crossover*.  
Antar induk yang telah ditentukan pada proses seleksi dilakukan kawin silang dengan *Partial Mapped Crossover* dengan menyilangkan tiga gen barisan pertama pada induk pertama dengan tiga gen barisan pertama pada induk kedua. Pada permasalahan vehicle routing problem pick-up and delivery termasuk pada jenis permasalahan permutasi yang memiliki batasan hanya melewati setiap kota sebanyak satu kali. Hal ini mengharuskan adanya penggantian gen yang memiliki nilai yang sama dengan gen yang tidak muncul setelah dilakukan proses *crossover*. Probabilitas terjadinya *crossover* yaitu 0.8 yang menandakan 80% dari total populasi kemungkinan akan melakukan *crossover*. Pada penentuan induk yang akan melakukan *crossover* yaitu dengan membangkitkan nilai random. [4]
- Mutasi.  
Melakukan proses mutasi dengan menukarkan urutan gen dalam satu kromosom. Proses ini adalah proses memunculkan kemungkinan solusi baru, namun diusahakan tidak terlalu banyak individu yang mengalami mutasi dengan menetapkan probabilitas mutasi 0.1. [4]
- Pemberhentian algoritma.  
Proses pembuatan populasi baru akan terus berulang hingga algoritma mendapatkan hasil yang sesuai dengan kriteria yang telah ditentukan. Selain itu

algoritma dapat berhenti sesuai dengan kondisi yang ditentukan. Kondisi ini salah satunya adalah banyaknya generasi. Algoritma akan berhenti ketika jumlah generasi mencapai nilai `max_iterasi` yang ditentukan di awal, misalkan 100. Dan kondisi lain yaitu apabila populasi yang dihasilkan konvergen maka proses iterasi selesai.

#### **4.3.2. Desain Algoritma Pencarian Tabu**

Untuk kromosom yang merupakan masukan dari algoritma pencarian tabu yaitu kromosom yang memiliki nilai fitness terbesar dari seluruh kromosom hasil seluruh generasi pada proses algoritma genetika. Algoritma pencarian tabu menggunakan tabu list untuk menyimpan sekumpulan solusi yang baru saja dievaluasi.

Pada setiap iterasi solusi akan dicocokkan terlebih dahulu dengan isi tabu list untuk melihat apakah solusi tersebut sudah ada pada tabu list, maka solusi tersebut tidak akan dievaluasi kembali pada iterasi berikutnya. Apabila sudah tidak ada lagi solusi yang tidak menjadi anggota tabu list, maka nilai terbaik yang baru saja diperoleh dengan total jarak terpendek merupakan solusi terbaik.

#### **4.3.3. *Hybrid GA-TS***

Metode yang digunakan pada tugas akhir ini adalah metode hybrid GA-TS. Peletakan TS berada setelah proses GA selesai dan menghasilkan output satu kromosom terbaik lalu dicek menggunakan TS.

Data yang digunakan merupakan data jarak studi kasus sebagai berikut.

**Tabel 4.1 Data Jarak (1)**

<b>Node</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>1</b>	0	18	11.2	8	28.5	33.4	26.7	42.2	19.5	14.7	30.2
<b>2</b>	18	0	14.2	21.5	23	35.2	19.9	48.4	26.8	22	37.5
<b>3</b>	11.20	13.1	0	14.6	19.7	27.7	16.5	38.1	20.5	15.1	27.2
<b>4</b>	8	21.5	14.6	0	33.2	39.7	30	48.4	26.8	22	37.5
<b>5</b>	28.5	23	19.7	33.2	0	9.8	5	35.1	24.7	28.7	31.4
<b>6</b>	33.4	35.2	27.8	39.8	9.9	0	14.4	36.9	30	34	38.5
<b>7</b>	26.7	19.9	16.5	30	5	14.3	0	37	24.5	26.7	31.2
<b>8</b>	42.2	49.2	38.9	49.2	35.1	36.8	37.2	0	22.9	27.9	14.1
<b>9</b>	19.5	25.8	20.5	26.8	24.9	29.8	24.5	22.1	0	5.4	11.1
<b>10</b>	14.7	22.9	15.9	22.9	26.3	31.2	27.5	27.1	5.4	0	16.2
<b>11</b>	30.2	38.3	28	38.3	32.4	38.3	32	14.1	12	17	0
<b>12</b>	11.6	18.5	11.5	18.5	21.8	26.7	21.4	31.7	10	6.1	20.8
<b>13</b>	8.8	16.8	9.9	16.8	22.8	27.8	20.6	34.7	13	8.3	23.8
<b>14</b>	43.5	59.7	52.8	47	46.5	75.5	56.9	66.7	14.3	42	55.3
<b>15</b>	6.4	18.3	11.3	14.6	26.5	31.5	24.3	34.7	13	8.2	23.8

<b>16</b>	21.1	26.2	19.2	26.2	31.8	36.8	31.4	28.2	6.6	4.8	17.3
<b>17</b>	34.8	49.7	42.8	41.7	55.4	60.3	55	40.4	30.1	28.4	34.2
<b>18</b>	54	51.7	48.2	60.2	34	25.2	38.5	47.5	47.7	54.4	49
<b>19</b>	54.2	51.9	48.4	60.4	34.2	25.4	38.7	47.7	47.9	54.6	49.2
<b>20</b>	45.2	35.6	38.8	48.6	24.5	16.1	26.3	51.5	44.6	48.6	53.1
<b>21</b>	55.9	52.4	50.2	62.2	35.9	27.2	40.4	49.4	49.7	56.3	51

Tabel 4.2 Data Jarak (2)

<b>Node</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>
<b>1</b>	11.6	8.8	45.3	6.4	21.1	34.8	54	54.1	45.2	55.9
<b>2</b>	19.3	25	55.7	18.5	26.2	49.6	51.7	51.8	35.6	52.4
<b>3</b>	12.3	8	49.4	11.6	37.4	42	48.8	48.9	41.5	50.6
<b>4</b>	19.3	15	47	14.6	32.4	41.6	59.3	59.4	52	61.2
<b>5</b>	21.6	20.7	69.3	26.5	53.7	34.6	34.1	34.2	44.8	50
<b>6</b>	26.9	25.9	74.4	31.7	58.8	60.9	25.3	25.4	23.2	32.5
<b>7</b>	21.4	18.6	56	24.4	53.4	42.5	39.1	39.2	35.5	40.9
<b>8</b>	32.1	35.3	65.9	35.5	49.3	43.4	46	47	48.7	47.9

<b>9</b>	9.6	13.5	12.9	13	28.6	29.6	48.8	48.9	48.8	50.7
<b>10</b>	5.7	9.6	41.1	9.1	26.8	28.9	50.6	50.7	48.6	52.5
<b>11</b>	21.2	24.4	57.6	24.6	39.4	33.1	48.8	48.9	51.4	50.6
<b>12</b>	0	4.4	47	6.3	31.4	33.5	47.2	47.2	43.8	49
<b>13</b>	4.4	0	98.4	6.6	34.4	36.5	47.6	47.6	44.3	49.5
<b>14</b>	47.5	48	0	42.2	47.8	44.9	92.6	92.7	92.5	94.4
<b>15</b>	7.2	6.8	42.3	0	26.7	30.1	51.8	51.9	48.5	53.7
<b>16</b>	9	12.9	37.6	12.4	0	24.1	54	55	53.9	55.9
<b>17</b>	32.5	36.4	45.2	30.6	29.7	0	77.8	77.9	80.1	82.8
<b>18</b>	47.3	46.3	90.6	52.2	75.1	80	0	1	12.9	1.3
<b>19</b>	47.5	46.5	92.6	52.4	77.1	82	1	0	14.9	3.3
<b>20</b>	41.5	40.5	87.5	45.6	73.6	79.1	9.9	9.1	0	14.6
<b>21</b>	49.2	48.2	93.5	54.1	77.9	82.8	2.3	2.4	14.5	0

## a. Gen

Melihat banyaknya node sebanyak 21 pada satu kromosom. Setiap node merepresentasikan lokasi pengiriman barang dari node tersebut sehingga kita dapat mengetahui lokasi dan jarak dari *node* yang akan dikunjungi. Data permintaan DAMIU Banyu Belik ada pada sub-bab 4.1.

## b. Kromosom

Berdasarkan jumlah *customers*, maka dapat disimpulkan panjang kromosom sebanyak 21 node. Karena kasus VRP-PD merupakan kasus pengiriman barang dari satu tempat dan nantinya kembali ke tempat tersebut maka pada kromosom awalan dan akhiran ada pada depot yaitu dengan direpresentasikan node 1, maka panjang node kromosom sebanyak 22 node. Berikut salah satu inisialisasi kromosom sepanjang 22 node.

1	4	6	7	3	8	20	13	15	2	12	19	21	16	14	9	10	11	5	18	17	1
---	---	---	---	---	---	----	----	----	---	----	----	----	----	----	---	----	----	---	----	----	---

## c. Populasi

Setelah membentuk kromosom lebih dari satu, maka kumpulan kromosom tersebut dinamakan populasi, selanjutnya akan melakukan iterasi melalui proses seleksi, crossover, dan mutasi hingga mencapai kriteria pemberhentian dan memberikan solusi satu kromosom terbaik dan dilanjutkan dengan proses pencarian tabu yang telah dijelaskan pada sub-bab 4.3.1 dan 4.3.2.

*Halaman ini sengaja dikosongkan*



## **BAB V IMPLEMENTASI**

Pada bab ini berisi tentang proses implementasi algoritma ke dalam *tool* matlab.

### **5.1. Implementasi Algoritma Genetika**

Seperti yang telah didesain pada bab sebelumnya, pada sub-bab ini menjelaskan implementasi algoritma genetika ke dalam *tool* matlab sesuai dengan proses yang telah dijelaskan pada sub-bab 4.3.1. Pada setiap sub-bab tahapan algoritma disertakan juga potongan kode yang dikemas dalam MFile, beserta penjelasannya.

#### **5.1.1. Inisialisasi Parameter**

Sintaks function pada code 5.1 di bawah mendeklarasikan bahwa M-File tersebut bernama “GATS” yang jika dijalankan membutuhkan variable input yaitu “dataVRPPD”, “jarak\_antar\_node”, “kapasitas\_kendaraan”, “pop\_size”, “mutrate”, “CR”, dan “max\_iterasi”. Fungsi ini menjadi inputan untuk menginisialisasikan parameter yang dipakai pada proses algoritma genetika.

```
function [solusi_akhir_tabu2  
tabu__iterasi]=GATS(dataVRPPD,  
jarak_antar_node,kapasitas_kendaraan,pop_size,  
mutrate,CR,max_iterasi)
```

#### **Code 5.1 Inisialisasi Parameter**

Pada code 5.2 terdapat beberapa variabel yang dihasilkan dari inputan awal dan dipakai untuk menginisialisasikan parameter yang terdapat pada GA. Pendefinisian dari data yang memerlukan pemanggilan yaitu dataVRPPD.m dan jarak\_antar\_node.m. Proses ini ditunjukkan pada code 5.2.

Variabel  $n$  dan  $m$  menunjukkan ukuran dari data yang ada di dalam dataVRPPD. Variabel “pengiriman” dan “ambilbarang” didapatkan pada kolom ke dua dan ke tiga dalam dataVRPPD.

### 5.1.2. Inisialisasi Populasi

```
[dataVRPPD]=dataVRPPD;
[n m]=size (dataVRPPD);
[jarak antar_node] = jarak antar_node;
pengiriman=dataVRPPD(:,2);
ambilbarang=dataVRPPD(:,3);
kapasitas=kapasitas_kendaraan;
jumlah_node=length(jarak antar_node);
t=cputime;
```

**Code 5.2 Variabel Inputan**

Pada tahap ini merupakan tahap untuk menghasilkan beberapa jumlah kromosom baru atau dinamakan dengan populasi. Pembuatan populasi dilakukan secara acak atau *random*. Pada code 5.3 dijelaskan, bahwa untuk menghasilkan sejumlah kromosom, maka dilakukan iterasi sebanyak `pop_size` pada masukan yang telah ditentukan di awal.

```
for k=1:pop_size;
    rand_numb=[1 rand(1,jumlah_node-1) 1];
    [urut_tengah
idx_urut]=sort(rand_numb(2:end-1));
    tambah_satu=ones(1,jumlah_node-1);
    calon_gen=idx_urut+tambah_satu;

    gen2=[1 calon_gen 1];
```

**Code 5.3 Inisialisasi Populasi**

### 5.1.3. Melakukan Perhitungan Nilai Fitness dan Elitisme

Langkah selanjutnya adalah melakukan penghitungan nilai fitness setiap kromosom. Kromosom dengan nilai fitness terbaik akan di kloning (kromosom dengan nilai fitness terendah merupakan kromosom dengan jarak paling kecil), kemudian kromosom tersebut menggantikan kromosom dengan nilai fitness terendah pertama dan kedua. Tujuan kloning adalah membesar kemungkinan terpilihnya kromosom dengan nilai tertinggi pada saat proses seleksi. Proses kloning tersebut dinamakan dengan *elitsm* dan dapat dilihat pada code 5.4.

```
[totaljarak routes
jumlah_hari]=Constraints_Evaluation(dataVRPPD,
jarak_antar_node,
kapasitas_kendaraan,populasi_awal);
    total_jarak_seleksiawal=totaljarak;

    [bea
pilih]=sort(total_jarak_seleksiawal, 'ascend');

    pop_sortir=populasi_awal(pilih,:);
    pop_urut=pop_sortir(1:pop_size,:);
    individu_terbaik=pop_sortir(1:1,:);
    f_average=mean(total_jarak_seleksiawal);
    f_max=min(bea);
    nomortertbaik=pilih(1 ,1);
    nomorterburuk=pilih(pop_size ,:);
    nomorterburuk2=pilih(pop_size-1 ,:);
```

**Code 5.4 Elitisme**

#### 5.1.4. Seleksi

Untuk proses seleksi, pada kasus ini, metode yang digunakan untuk seleksi adalah *roulette wheel*. [4] Nilai  $r$  yang dibangkitkan adalah 100, artinya, kromosom yang terpilih pada seleksi ini adalah sebanyak 100 kromosom. Pada code 5.5 merupakan code yang dipakai pada proses ini.

#### 5.1.5. Crossover

```
for iterasii=1:pop_size

    fitnessindividu=total_jarak_seleksiawal(iterasii)/totalfitness;

    totalfitnessindividu=[totalfitnessindividu;fitnessindividu];
end

%menghitung nilai fitness komulatif tiap individu
for iterasiii=1:pop_size
    if(iterasiii==1)
        fitnesskomulatif=fitnessindividu(1,1);

    totalfitnesskomulatif=[totalfitnesskomulatif;fitnesskomulatif];
    else
        y=iterasiii-1;

    fitnesskomulatif=totalfitnessindividu(iterasiii,1)+totalfitnesskomulatif(y,1);

    totalfitnesskomulatif=[totalfitnesskomulatif;fitnesskomulatif];
end
```

**Code 5.5 Seleksi**

Pada tahap ini merupakan tahap yang dilakukan untuk memilih *parent* dengan cara melakukan perkawinan silang terhadap

*parent* yang berbeda atau dapat diilustrasikan pada gambar 5.1. Banyaknya jumlah *parent* yang di *crossover* ditentukan oleh besarnya peluang *crossover* dan membangkitkan nilai random.

<i>Parent 1</i>	A1	A2
<i>Parent 2</i>	B1	B2
Hasil <i>Crossover</i>		
<i>Offspring 1</i>	A1	B2
<i>Offspring 2</i>	B1	A2

**Gambar 5.1 Crossover**

Pemilihan *parent* dilakukan secara acak sejumlah *parent* dari hasil seleksi *roulette wheel*. Apabila *parent* terpilih keduanya sama, maka akan dilakukan pengacakan *parent* ulang. Nilai  $r$  yang dibangkitkan dibandingkan dengan peluang *crossover* ( $P_c$ ). Apabila nilai  $r < P_c$ , maka kromosom tersebut terpilih menjadi *parent*.

Pada code 5.5 dijelaskan bahwa tiga gen pertama pada *parent* pertama akan ditukar dengan tiga gen pada *parent* kedua. Lalu gen yang sama dari hasil perkawinan silang atau disebut *offspring*, ditukar dengan gen yang hilang, dengan urutan sama seperti pada *parent* pertama. Begitupula untuk *offspring* kedua, proses yang dilakukan sama.

```

for cross=1:jmlh_iterasi_cross%memulai iterasi
    parent1=1+parent2;
    parent2=parent1+1;

    %offspring 1%
    aa=[kumpulanparent2(parent1,(1:jum_gen))]; %awalgen parent1
    bb=[kumpulanparent2(parent2,(jum_gen+1:(kk)))]; %akhirgen parent2
    cc=[kumpulanparent2(parent2,(1:jum_gen))]; %awalgen parent2
    x1 = aa([1],[1]); %mengambil nilai gen pertama parent1
    x2 = aa([1],[2]); %mengambil nilai gen kedua parent1
    x3 = aa([1],[3]); %mengambil nilai gen ketiga parent1
    cc1 = cc([1],[1]); %mengambil nilai gen pertama parent2
    cc2 = cc([1],[2]); %mengambil nilai gen kedua parent2
    cc3 = cc([1],[3]); %mengambil nilai gen ketiga parent2
    for cek=1:1
        bb(bb==x1) = cc1; %me replace angka yang sama dengan angka yang hilang
        bb(bb==x2) = cc2; %me replace angka yang sama dengan angka yang hilang
        bb(bb==x3) = cc3; %me replace angka yang sama dengan angka yang hilang
        hasil_cross = [1 aa bb 1];
        hasil_cross4 = [aa bb]; %hasil crossover offspring1
    end
end

```

### Code 5.6 Crossover

#### 5.1.6. Mutasi

Pada tahap ini akan dilakukan pertukaran gen pada satu kromosom yang sama. Kromosom yang dimaksud adalah kromosom hasil perbandingan nilai random pada setiap kromosom dengan peluang mutasi ( $P_m$ ). Jika nilai random pada kromosom  $n < P_m$ , maka kromosom tersebut mengalami mutasi. Pada code 5.7 merupakan sintaks untuk melakukan mutasi.

```

for icc=1:n
    ix=icc;
    iy=ceil(m*rand); %merandom gen beberapa yang akan dir
    iy2=ceil(m*rand); %merandom gen beberapa yang akan dir

    mut1=kumpulparentmut(ix,iy); %mengambil nilai pada n
    mut2=kumpulparentmut(ix,iy2); %mengambil nilai pada
    kk=kumpulparentmut(ix,:); %kk adalah kromosom yang al
    kk(1,iy)=mut2; %pertukaran nilai pada kromosom dilak
    kk(1,iy2)=mut1; %pertukaran nilai pada kromosom dilak
    |
    ambilkolommut=kmplnomermut(n,:);
    populasi_setelah_crossover(ambilkolommut,:)=kk(1,:);
end
end

```

Code 5.7 Mutasi

## 5.2. Kriteria Pemberhentian

Terdapat dua kriteria pemberhentian pada penelitian ini. Yang pertama adalah apabila standar deviasi dari total nilai fitness dari semua kromosom hasil dari setiap generasi yang terbentuk adalah konvergen. Nilai yang digunakan untuk menyatakan konvergen menggunakan standar deviasi yaitu sepuluh. Yang kedua menggunakan max\_iterasi. Pada kriteria pemberhentian pertama digunakan code sebagai berikut:

```

totstd=sum(std(total_jarak_konvergen));
if (totstd < 10)
    iterasi=iterasi+1000;
end

```

Code 5.8 Konvergen dan Maksimum Iterasi

## 5.3. Implementasi Algoritma Pencarian Tabu

Pada sub-bab ini dijelaskan implementasi dari sebuah proses yang dinamakan *tabu search*. Kromosom terbaik yang

merupakan hasil dari optimasi algoritma genetika akan dilanjutkan ke dalam proses ini. Untuk melakukan *tabu search*, proses yang pertama adalah menentukan jumlah kromosom dalam populasi *tabu search*, populasi yang dimaksud berisi kromosom-kromosom yang memenuhi kriteria aspirasi, proses yang kedua adalah menentukan kriteria aspirasi, yaitu nilai yang diharapkan, dan yang terakhir adalah *tabu tenure*. Jumlah dari *tabu tenure* menentukan bahwa atribut yang terdapat dalam *tabu list* tidak boleh dipakai sebanyak jumlah *tabu tenure* tersebut. Misalnya, *tabu tenure* bernilai dua yang artinya atribut pada *tabu list* tidak boleh muncul selama dua kali iterasi selanjutnya, contohnya atribut yang termasuk *tabu list* adalah 3 dan 1. Maka atribut 3 dan 1 tidak boleh terpilih selama dua iterasi kedepan. Konsep perpindahan *node* pada *tabu search* sama seperti pada tahap mutasi. Hasil kromosom dari *tabu search* yang memenuhi kriteria aspirasi, akan dimasukkan ke dalam populasi pencarian *tabu* dan seterusnya sampai dengan populasi pencarian *tabu* tercapai. Nilai awal dari kriteria aspirasi merupakan hasil dari nilai fitness algoritma genetika, kemudian apabila terdapat kromosom yang memiliki nilai fitness lebih baik dari kriteria aspirasi sebelumnya, maka nilai kriteria aspirasi diperbarui dengan nilai fitness tersebut.

Untuk tahap inisialisasi *tabu search* dapat dilihat pada code 5.9. *Input* yang terdapat dalam tahap inisialisasi tersebut menentukan jalannya sebuah proses *tabu search*. Seperti maksimal kromosom dalam satu populasi dan lain lain.



```

%TABU SEARCH
%=====
nilai_diharapkan=jarakakhirGA;
max_tabu_iterasi=2000; %max ite
jarak=2000; |
total_move=5;
kromosom_move=[]; %hasil TS yg
izz=0; %variabel untuk menyimpan
izz=1; %variabel untuk menyimpa
tabu1=0; %variabel untuk menyim
tabu2=0;
tabu3=0;
tabu__iterasi=1;
tabutenur1=0;
tabutenur2=0;
tabutenur3=0;
populasi_TS=400;

```

#### Code 5.9 Input Pencarian Tabu

Untuk proses *tabu search*, setiap satu kali iterasi akan dilakukan lima kali *move* atau perpindahan node. Kromosom hasil *move* terbaik akan dilanjutkan pada iterasi selanjutnya dan seterusnya, lalu setiap kromosom hasil *move* terbaik memenuhi kriteria aspirasi, maka kromosom tersebut diambil dan dimasukkan kedalam populasi hasil *tabu search* dan seterusnya sampai dengan total kromosom memenuhi populasi maksimum yang diinginkan.

Pada code 5.10 merupakan contoh dari salah satu iterasi, dimana dalam iterasi tersebut terdapat atribut dalam tabu list. Variabel abcd, abcd2, efgh, efgh2, ijkl, dan ijkl2 merupakan atribut yang termasuk dalam tabu list. Apabila terdapat atribut yang tabu pada iterasi berikutnya, maka kromosom tersebut tidak dipilih.

```

%=====Move 1=====
tabu_iterasi=tabu_iterasi+1;
iz=ceil(18*rand+1); %merandom kolom1 yang akan di move
izz=ceil(18*rand+1); %merandom kolom2 yang akan di move
kolommove=[iz izz];
kolommove2=[iz izz];
if(kolommove==abcd & tabutenur1>0 | kolommove==abcd2 & tabutenur1>0)
    iz=ceil(18*rand+1); %merandom kolom1 yang akan di move
    izz=ceil(18*rand+1); %merandom kolom2 yang akan di move
end
if(kolommove==efgh & tabutenur2>0 | kolommove==efgh2 & tabutenur2>0)
    iz=ceil(18*rand+1); %merandom kolom1 yang akan di move
    izz=ceil(18*rand+1); %merandom kolom2 yang akan di move
end
if(kolommove==ijkl & tabutenur3>0 | kolommove==ijkl2 & tabutenur3>0)
    iz=ceil(18*rand+1); %merandom kolom1 yang akan di move
    izz=ceil(18*rand+1); %merandom kolom2 yang akan di move
end
move=terpilih(1,iz); %mengambil nilai yang akan di move
move2=terpilih(1,izz); %mengambil nilai yang akan di move
k1=terpilih(1,:);
k1(1,izz)=move; %melakukan move
k1(1,iz)=move2; %melakukan move

```

**Code 5.10 Move pada satu Iterasi**

#### 5.4. Nilai Fitness Akhir

Pada tahap ini terdapat kromosom terbaik yang terpilih dalam proses GA-TS, kemudian dilakukan penghitungan ulang terhadap jarak rute yang ditempuh dari kromosom tersebut, kemudian ditampilkan hasilnya dalam *command window*. Apabila hasil dari algoritma pencarian tabu lebih baik daripada hasil akhir dari algoritma genetika, maka hasil algoritma pencarian tabu menggantikan hasil akhir algoritma genetika sebagai rute optimal untuk memecahkan masalah VRP-PD. Code 5.11 menjelaskan cara menghitung total jarak pada kromosom yang terpilih.

```

-
Final Test
=====

populasi_akhir=[solusi_final];
[total_jarak routes jumlah_hari]=Constraints_Evaluation(dataVRPFD, jarak_antar_node, kapasitas_kendaraan, populasi_akhir);
total_jarak_parents=total_jarak;
[jarak_terminimum idx]=min(total_jarak_parents);
solusi_akhir=populasi_akhir(idx,:);

```

### Code 5.11 Menghitung Total Jarak

*Halaman ini sengaja dikosongkan*

## **BAB VI**

### **HASIL DAN PEMBAHASAN**

Pada bab ini menjelaskan mengenai proses uji coba dan analisis terhadap hasil yang diperoleh dari proses implementasi kombinasi algoritma genetika dan pencarian tabu.

#### **6.1. Lingkungan Uji Coba**

Lingkungan uji coba membahas mengenai lingkungan pengujian yang digunakan untuk implementasi tugas akhir ini meliputi perangkat keras dan perangkat lunak yang digunakan. Spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam implementasi algoritma genetika ditunjukkan pada Tabel 6-1.

**Tabel 6.1 Lingkungan Uji Coba Perangkat Keras**

<b>Perangkat Keras</b>	<b>Spesifikasi</b>
<b>Jenis</b>	Notebook HP
<b>Processor</b>	Intel Core i5
<b>RAM</b>	4 GB
<b>Hard Disk Drive</b>	500 GB

Lingkungan perangkat lunak yang digunakan dalam uji coba model ditunjukkan pada Tabel 6-2.

**Tabel 6.2 Lingkungan Uji Coba Perangkat Lunak**

<b>Perangkat Lunak</b>	<b>Fungsi</b>
<b>Windows 8.1</b>	Sistem Operasi
<b>Matlab R2014a</b>	Membuat model matematis

Perangkat Lunak	Fungsi
	Melakukan optimasi
<b>Microsoft Excel 2013</b>	Mengolah data Melakukan validasi Pembuatan grafik node

## 6.2. Parameter Uji Coba

Penggunaan parameter sebagai inputan pada proses algoritma genetika dan GA-TS dilakukan untuk mengetahui bagaimana kombinasi-kombinasi dari parameter yang dapat menghasilkan total keuntungan yang paling tinggi. Hasil yang diperoleh dari algoritma genetika dan GA-TS merupakan bilangan acak sehingga uji coba akan dilakukan lebih dari satu kali. Parameter-parameter yang digunakan sebagai perbandingan hasil algoritma genetika dan GA-TS pada tugas akhir ini adalah sebagai berikut.

- Perbandingan pengaruh banyaknya populasi ( $N_{pop}$ ) pada total jarak.
- Perbandingan pengaruh nilai probabilitas pindah silang ( $p_c$ ) pada total jarak.
- Perbandingan pengaruh nilai probabilitas mutasi ( $P_m$ ) dan penggunaan fungsi mutasi pada total jarak.
- Perbandingan pengaruh ada tidaknya kombinasi pencarian tabu pada total jarak.

## 6.3. Skenario Uji Coba

Pada pengerjaan tugas akhir ini dibutuhkan skenario uji coba untuk mengetahui hasil algoritma genetika dan GA-TS yang paling optimal. Skenario ini nantinya dilakukan menggunakan dua data, yaitu dataset dari Solomon untuk mengetahui kemampuan algoritma genetika dan GA-TS dalam jumlah besar dan memberikan output yang benar berdasarkan penelitian

sebelumnya “*A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem*” [5] dan data perusahaan yang nantinya akan berpengaruh pada total jarak dari banyaknya kota yang harus dikunjungi oleh pengirim barang dari Depot Air Minum Kemasan Galon Banyu Belik untuk memberikan rekomendasi rute pengiriman barang untuk perusahaan. Uji coba yang dicatat dalam tugas ini menggunakan maksimum iterasi 100. Beberapa skenario tersebut antara lain sebagai berikut.

- a. Perbandingan total jarak berdasarkan parameter  $N_{pop}$  sebanyak 100, 200 dan 300 pada setiap percobaan.
- b. Perbandingan total jarak berdasarkan parameter  $P_c$  antara 0.2 – 0.9 pada setiap percobaan.
- c. Perbandingan total jarak berdasarkan parameter  $P_m$  yaitu 0,05 dan 0,1 pada setiap percobaan.
- d. Perbandingan pengaruh penggunaan kombinasi algoritma pencarian tabu pada setiap percobaan.

#### **6.4. Hasil Uji Coba Dataset**

Pembahasan hasil uji coba akan dilakukan dengan melakukan perbandingan dari total jarak yang didapat dari seluruh percobaan. Uji coba dilakukan untuk mengetahui jarak terpendek dengan mencoba parameter yang berbeda-beda. Berikut merupakan hasil uji coba yang dilakukan menggunakan dataset Solomon [5].

##### **6.4.1. Hasil Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300 , $P_m$ 0.05 , dan $P_c$ 0.2-0.9**

Uji Coba pada bagian ini dilakukan dengan menggunakan parameter  $N_{pop}$  sebanyak 100, 200, dan 300, tanpa mengkombinasi algoritma genetika dengan pencarian tabu. Dengan menggunakan  $P_c = 0.2-0.9$  dan  $P_m = 0.05$ . Berikut merupakan hasil uji coba pada perbandingan total keuntungan berdasarkan parameter  $N_{pop}$ . Tabel 6.3 menunjukkan hasil uji coba algoritma genetika dengan

menggunakan parameter  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.3 Uji Coba  $N_{pop} = 100$ ,  $P_m = 0.05$**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak GA</b>
0.2	0.05	100	1922.5
0.3	0.05	100	2051.7
0.4	0.05	100	1970.5
0.5	0.05	100	1901.4
0.6	0.05	100	1957.5
0.7	0.05	100	1959.2
0.8	0.05	100	2021.4
0.9	0.05	100	1926.2

Tabel 6.4 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.4 Uji Coba  $N_{pop} = 200$ ,  $P_m = 0.05$**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak GA</b>
0.2	0.05	200	1986.2
0.3	0.05	200	2084.3
0.4	0.05	200	1991.5
0.5	0.05	200	2026.2
0.6	0.05	200	2018.5
0.7	0.05	200	1887.3
0.8	0.05	200	1948.3
0.9	0.05	200	2007.3



Tabel 6.5 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.5 Uji Coba  $N_{pop} = 300$ ,  $P_m = 0.05$**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak GA</b>
0.2	0.05	300	1957.1
0.3	0.05	300	2021.4
0.4	0.05	300	1824.2
0.5	0.05	300	1793.4
0.6	0.05	300	1867.6
0.7	0.05	300	1881.2
0.8	0.05	300	1795.01
0.9	0.05	300	1969.1

#### **6.4.2. Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300 , $P_m$ 0.1 , dan $P_c$ 0.2-0.9.**

Tabel 6.6 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.6 Uji Coba  $N_{pop}=100$ ,  $P_c=0.2-0.9$ ,  $P_m=0.1$**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak GA</b>
0.2	0.1	100	1894.4
0.3	0.1	100	1966.4
0.4	0.1	100	1931.1
0.5	0.1	100	1875.7

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak GA</b>
0.6	0.1	100	1925.4
0.7	0.1	100	1813.2
0.8	0.1	100	1909.9
0.9	0.1	100	1930.2

Tabel 6.7 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.7 Uji Coba  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak GA</b>
0.2	0.1	200	2071.4
0.3	0.1	200	1869.5
0.4	0.1	200	1920.8
0.5	0.1	200	1793.9
0.6	0.1	200	1853.4
0.7	0.1	200	1852.5
0.8	0.1	200	1927.4
0.9	0.1	200	1924.7

Tabel 6.8 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.8 Uji Coba  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak GA</b>
0.2	0.1	300	1993
0.3	0.1	300	1963.2

Pc	Pm	Npop	Total Jarak GA
0.4	0.1	300	1990.6
0.5	0.1	300	1933.2
0.6	0.1	300	1863.4
0.7	0.1	300	1952.2
0.8	0.1	300	1847.2
0.9	0.1	300	1969.5

**6.4.3. Percobaan Menggunakan Algoritma Genetika dengan Parameter  $N_{pop}$  100, 200, 300,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$ , dan Kombinasi Algoritma Pencarian Tabu**

Tabel 6.9 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$ .

**Tabel 6.9 Uji Coba  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$**

Pc	Pm	Npop	GA-TS
0.2	0.05	100	1821.5
0.3	0.05	100	1902.7
0.4	0.05	100	1835.5
0.5	0.05	100	1780.4
0.6	0.05	100	1853.5
0.7	0.05	100	1845.2
0.8	0.05	100	1894.4
0.9	0.05	100	1824.2

Tabel 6.10 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$ .

**Tabel 6.10 Uji Coba  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$** 

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>GA-TS</b>
0.2	0.05	200	1838.2
0.3	0.05	200	1942.3
0.4	0.05	200	1842.5
0.5	0.05	200	1890.2
0.6	0.05	200	1901.5
0.7	0.05	200	1772.3
0.8	0.05	200	1832.3
0.9	0.05	200	1888.3

Tabel 6.11 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$ .

**Tabel 6.11 Uji Coba  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$** 

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>GA-TS</b>
0.2	0.05	300	1812.1
0.3	0.05	300	1879.4
0.4	0.05	300	1698.2
0.5	0.05	300	1692.4
0.6	0.05	300	1724.6
0.7	0.05	300	1740.2
0.8	0.05	300	1679.01
0.9	0.05	300	1823.1

#### 6.4.4. Percobaan Menggunakan Algoritma Genetika dengan Parameter Npop 100, 200, 300, $P_c = 0.2-0.9$ , $P_m = 0.1$ , dan Kombinasi Algoritma Pencarian Tabu

Tabel 6.12 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$ .

**Tabel 6.12 Uji Coba  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>GA-TS</b>
0.2	0.1	100	1768.4
0.3	0.1	100	1823.4
0.4	0.1	100	1830.1
0.5	0.1	100	1742.7
0.6	0.1	100	1804.4
0.7	0.1	100	1682.2
0.8	0.1	100	1790.9
0.9	0.1	100	1802.2

Tabel 6.13 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$

**Tabel 6.13 Uji Coba  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>GA-TS</b>
0.2	0.1	200	1922.4
0.3	0.1	200	1755.5
0.4	0.1	200	1812.8
0.5	0.1	200	1690.9
0.6	0.1	200	1718.4
0.7	0.1	200	1722.5

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>GA-TS</b>
0.8	0.1	200	1799.4
0.9	0.1	200	1813.7

Tabel 6.14 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$

**Tabel 6.14 Uji Coba  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>GA-TS</b>
0.2	0.1	300	1864
0.3	0.1	300	1822.2
0.4	0.1	300	1876.6
0.5	0.1	300	1822.2
0.6	0.1	300	1734.4
0.7	0.1	300	1811.2
0.8	0.1	300	1723.2
0.9	0.1	300	1845.5

### 6.5. Hasil Uji Coba Data Perusahaan

Pembahasan hasil uji coba akan dilakukan dengan melakukan perbandingan dari total jarak yang didapat dari seluruh percobaan. Uji coba dilakukan untuk mengetahui jarak terpendek dengan mencoba parameter yang berbeda-beda. Berikut merupakan hasil uji coba yang dilakukan menggunakan data Depot Banyu Belik.

### 6.5.1. Hasil Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300 , $P_m$ 0.05 , dan $P_c$ 0.2-0.9

Uji Coba pada bagian ini dilakukan dengan menggunakan parameter  $N_{pop}$  sebanyak 100, 200, dan 300, tanpa mengkombinasi algoritma genetika dengan pencarian tabu. Dengan menggunakan  $P_c = 0.2-0.9$  dan  $P_m = 0.05$ . Berikut merupakan hasil uji coba pada perbandingan total keuntungan berdasarkan parameter  $N_{pop}$ . Tabel 6.15 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.15 Uji Coba  $N_{pop} = 100$ ,  $P_m = 0.05$**

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA
0.2	0.05	100	786.7
0.3	0.05	100	807.7
0.4	0.05	100	748.6
0.5	0.05	100	734.5
0.6	0.05	100	776.1
0.7	0.05	100	767.3
0.8	0.05	100	778.1
0.9	0.05	100	745.6

Tabel 6.16 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.16 Uji Coba  $N_{pop} = 200$ ,  $P_m = 0.05$** 

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA
<b>0.2</b>	0.05	200	801.1
<b>0.3</b>	0.05	200	782.9
<b>0.4</b>	0.05	200	758.5
<b>0.5</b>	0.05	200	763.5
<b>0.6</b>	0.05	200	783.2
<b>0.7</b>	0.05	200	793.3
<b>0.8</b>	0.05	200	754
<b>0.9</b>	0.05	200	780

Tabel 6.17 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.17 Uji Coba  $N_{pop} = 300$ ,  $P_m = 0.05$** 

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA
<b>0.2</b>	0.05	300	799.3
<b>0.3</b>	0.05	300	782.5
<b>0.4</b>	0.05	300	776.8
<b>0.5</b>	0.05	300	779.9



$P_c$	$P_m$	$N_{pop}$	Total Jarak GA
<b>0.6</b>	0.05	300	767.1
<b>0.7</b>	0.05	300	748.4
<b>0.8</b>	0.05	300	774.3
<b>0.9</b>	0.05	300	796.4

### 6.5.2. Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300 , $P_m$ 0.1 , dan $P_c$ 0.2-0.9.

Tabel 6.18 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.18 Uji Coba  $N_{pop}=100$ ,  $P_c=0.2-0.9$ ,  $P_m=0.1$**

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA
<b>0.2</b>	0.1	100	790.5
<b>0.3</b>	0.1	100	774.9
<b>0.4</b>	0.1	100	787.2
<b>0.5</b>	0.1	100	779.7
<b>0.6</b>	0.1	100	757.9
<b>0.7</b>	0.1	100	731.1
<b>0.8</b>	0.1	100	754.2

<b>P<sub>c</sub></b>	<b>P<sub>m</sub></b>	<b>N<sub>pop</sub></b>	<b>Total Jarak GA</b>
<b>0.9</b>	0.1	100	785.1

Tabel 6.19 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.19 Uji Coba  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$**

<b>P<sub>c</sub></b>	<b>P<sub>m</sub></b>	<b>N<sub>pop</sub></b>	<b>Total Jarak GA</b>
<b>0.2</b>	0.1	200	790.8
<b>0.3</b>	0.1	200	763.7
<b>0.4</b>	0.1	200	771.7
<b>0.5</b>	0.1	200	758.8
<b>0.6</b>	0.1	200	773.8
<b>0.7</b>	0.1	200	786.7
<b>0.8</b>	0.1	200	757.6
<b>0.9</b>	0.1	200	788.6

Tabel 6.20 menunjukkan hasil uji coba algoritma genetika dengan menggunakan parameter  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$  tanpa mengkombinasikan dengan algoritma pencarian tabu.

**Tabel 6.20 Uji Coba  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$** 

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA
<b>0.2</b>	0.1	300	773.4
<b>0.3</b>	0.1	300	789
<b>0.4</b>	0.1	300	748.2
<b>0.5</b>	0.1	300	797.9
<b>0.6</b>	0.1	300	737.1
<b>0.7</b>	0.1	300	778.2
<b>0.8</b>	0.1	300	776.6
<b>0.9</b>	0.1	300	795.2

### **6.5.3. Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300, $P_c = 0.2-0.9$ , $P_m = 0.05$ , dan Kombinasi Algoritma Pencarian Tabu**

Tabel 6.21 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$ .

**Tabel 6.21 Uji Coba  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$** 

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA - TS
<b>0.2</b>	0.05	100	694.8

<b>P<sub>c</sub></b>	<b>P<sub>m</sub></b>	<b>N<sub>pop</sub></b>	<b>Total Jarak GA - TS</b>
<b>0.3</b>	0.05	100	730.5
<b>0.4</b>	0.05	100	695.5
<b>0.5</b>	0.05	100	692.7
<b>0.6</b>	0.05	100	697.6
<b>0.7</b>	0.05	100	697.5
<b>0.8</b>	0.05	100	715.1
<b>0.9</b>	0.05	100	696.7

Tabel 6.22 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$ .

**Tabel 6.22 Uji Coba Npop = 200, Pc = 0.2-0.9, Pm = 0.05**

<b>P<sub>c</sub></b>	<b>P<sub>m</sub></b>	<b>N<sub>pop</sub></b>	<b>Total Jarak GA - TS</b>
<b>0.2</b>	0.05	200	697.2
<b>0.3</b>	0.05	200	741
<b>0.4</b>	0.05	200	703.8
<b>0.5</b>	0.05	200	722.8
<b>0.6</b>	0.05	200	729.9
<b>0.7</b>	0.05	200	693.2
<b>0.8</b>	0.05	200	700.3

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA - TS
<b>0.9</b>	0.05	200	716.3

Tabel 6.23 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$ .

**Tabel 6.23 Uji Coba  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.05$**

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA - TS
<b>0.2</b>	0.05	300	693.1
<b>0.3</b>	0.05	300	716.8
<b>0.4</b>	0.05	300	689.9
<b>0.5</b>	0.05	300	688.9
<b>0.6</b>	0.05	300	694.1
<b>0.7</b>	0.05	300	694.5
<b>0.8</b>	0.05	300	689.7
<b>0.9</b>	0.05	300	705.1

#### 6.5.4. Percobaan Menggunakan Algoritma Genetika dengan Parameter $N_{pop}$ 100, 200, 300, $P_c = 0.2-0.9$ , $P_m = 0.1$ , dan Kombinasi Algoritma Pencarian Tabu

Tabel 6.24 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$ .

**Tabel 6.24 Uji Coba  $N_{pop} = 100$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$**

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA - TS
<b>0.2</b>	0.1	100	699.9
<b>0.3</b>	0.1	100	704.1
<b>0.4</b>	0.1	100	706.2
<b>0.5</b>	0.1	100	695.3
<b>0.6</b>	0.1	100	701.9
<b>0.7</b>	0.1	100	688.9
<b>0.8</b>	0.1	100	701
<b>0.9</b>	0.1	100	697.9

Tabel 6.25 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$

**Tabel 6.25 Uji Coba  $N_{pop} = 200$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$**

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA - TS
<b>0.2</b>	0.1	200	730.6

<b>P<sub>c</sub></b>	<b>P<sub>m</sub></b>	<b>N<sub>pop</sub></b>	<b>Total Jarak GA - TS</b>
<b>0.3</b>	0.1	200	696.3
<b>0.4</b>	0.1	200	702.7
<b>0.5</b>	0.1	200	694.9
<b>0.6</b>	0.1	200	698.8
<b>0.7</b>	0.1	200	698.8
<b>0.8</b>	0.1	200	688.8
<b>0.9</b>	0.1	200	700.9

Tabel 6.26 menunjukkan hasil uji coba kombinasi algoritma genetika dan pencarian tabu menggunakan parameter  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$

**Tabel 6.26 Uji Coba  $N_{pop} = 300$ ,  $P_c = 0.2-0.9$ ,  $P_m = 0.1$**

<b>P<sub>c</sub></b>	<b>P<sub>m</sub></b>	<b>N<sub>pop</sub></b>	<b>Total Jarak GA - TS</b>
<b>0.2</b>	0.1	300	712.4
<b>0.3</b>	0.1	300	714.3
<b>0.4</b>	0.1	300	715.1
<b>0.5</b>	0.1	300	705.9
<b>0.6</b>	0.1	300	695.8
<b>0.7</b>	0.1	300	700.5
<b>0.8</b>	0.1	300	695.4

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA - TS
<b>0.9</b>	0.1	300	712.2

### 6.5.5. Bagan Skenario

No	Skenario	Hasil	Algoritma	Rute
<b>1</b>	$P_c = 0.5$	734.5	GA	1-13-7-21-19-8- 16-9-14-5-3-12- 15-11-4-6-10- 20-18-2-17-1
	$P_m = 0.05$			
	$N_{pop} = 100$			
<b>2</b>	$P_c = 0.8$	754	GA	1-20-7-2-18-16- 6-9-14-4-3-12- 15-11-8-5-10- 21-19-13-17-1
	$P_m = 0.05$			
	$N_{pop} = 200$			
<b>3</b>	$P_c = 0.7$	748.4	GA	1-13-7-20-18-5- 6-9-14-4-3-12- 15-10-8-16-11- 21-19-2-17-1
	$P_m = 0.05$			
	$N_{pop} = 300$			
<b>4</b>	$P_c = 0.7$	731.1	GA	1-17-8-20-15-5- 6-9-14-4-3-12- 19-11-7-16-10- 21-18-13-2-1
	$P_m = 0.1$			
	$N_{pop} = 100$			
<b>5</b>	$P_c = 0.8$	757.6	GA	1-21-7-20-19- 17-6-9-14-4-3- 12-15-11-8-16- 10-2-18-13-5-1
	$P_m = 0.1$			
	$N_{pop} = 200$			
<b>6</b>	$P_c = 0.6$	737.1	GA	1-16-7-11-19-5- 6-9-14-4-3-12- 15-20-17-2-10- 21-18-13-8-1
	$P_m = 0.1$			
	$N_{pop} = 300$			
<b>7</b>	$P_c = 0.5$	692.7	GA-TS	



	P <sub>m</sub> = 0.05			1-15-7-20-19-8-6-9-14-17-3-12-2-11-5-16-10-21-18-13-4-1
	N <sub>pop</sub> = 100			
8	P <sub>c</sub> = 0.7	693.2	GA-TS	1-3-7-20-17-5-6-9-14-4-2-13-15-11-8-16-10-21-18-12-19-1
	P <sub>m</sub> = 0.05			
	N <sub>pop</sub> = 200			
9	P <sub>c</sub> = 0.5	688.9	GA-TS	1-8-7-20-19-17-6-9-14-4-3-12-15-11-2-16-10-21-18-13-5-1
	P <sub>m</sub> = 0.05			
	N <sub>pop</sub> = 300			
10	P <sub>c</sub> = 0.7	688.9	GA-TS	1-11-7-20-19-5-6-9-14-17-3-12-15-2-8-16-10-21-18-13-4-1
	P <sub>m</sub> = 0.1			
	N <sub>pop</sub> = 100			
11	P <sub>c</sub> = 0.8	688.8	GA-TS	1-2-7-20-19-5-6-9-14-4-3-12-15-11-8-16-10-21-18-13-17-1
	P <sub>m</sub> = 0.1			
	N <sub>pop</sub> = 200			

## 6.6. Analisa Hasil Uji Coba Dataset

Analisis hasil berisi penjelasan tentang semua hasil output yang didapat dari uji coba yang dilakukan dengan input parameter yang berbeda-beda.

### 6.6.1. Analisa Hasil Uji Coba Algoritma Genetika

Dari hasil uji coba algoritma genetika terhadap dataset RC105 Solomon [5] yang diujikan dalam parameter yang berbeda-beda didapatkan beberapa rute dengan total jarak yang berbeda.

Berdasarkan hasil dari Tabel 6.3, Tabel 6.4 dan Tabel 6.5 yaitu uji coba menggunakan parameter  $P_c=0.2-0.9$ ,  $P_m=0.05$  dan  $N_{pop}=100, 200$  dan  $300$  diperoleh total jarak terpendek pada

setiap tabel yaitu pada Tabel 6.3 dengan  $P_c = 0.9$ , dengan jarak 1926.2 km; pada Tabel 6.4 dengan  $P_c = 0.7$ , dengan jarak 1887.3 km dan pada Tabel 6.5 dengan  $P_c = 0.5$ , dengan jarak 1793.4 km. Hal ini menunjukkan bahwa jarak terendah berada pada Tabel 6.5 dengan nilai  $P_c = 0.5$  dengan jarak 1793.4 km.

Berdasarkan hasil dari Tabel 6.6, Tabel 6.7 dan Tabel 6.8 yaitu uji coba menggunakan parameter  $P_c$  0.2-0.9,  $P_m$  0.1 dan  $N_{pop}$  100, 200 dan 300 diperoleh total jarak terpendek pada setiap tabel yaitu pada Tabel 6.6 dengan  $P_c = 0.7$ , dengan jarak 1813.2 km; pada Tabel 6.7 dengan  $P_c = 0.5$ , dengan jarak 1793.9 km dan pada Tabel 6.8 dengan  $P_c = 0.8$ , dengan jarak 1847.2 km. Hal ini menunjukkan bahwa jarak terendah berada pada Tabel 6.7 dengan nilai  $P_c = 0.5$  dengan jarak 1793.9 km. Tabel 6.27 merupakan perbandingan hasil terbaik dari masing-masing uji coba algoritma dengan parameter yang berbeda-beda. Didapatkan hasil bahwa jarak terbaik menggunakan algoritma genetika adalah pada saat  $P_c = 0.5$ ,  $P_m = 0.05$  dan  $N_{pop}$  300 yaitu dengan total jarak 1793.4 km.

**Tabel 6.27 Hasil Terbaik Dari Masing-Masing Uji Coba Algoritma Genetika**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak</b>
0.9	0.05	100	1926.2
0.7	0.05	200	1887.3
0.5	0.05	300	1793.4
0.1	0.1	100	1813.2
0.5	0.1	200	1793.9
0.8	0.1	300	1847.2

### **6.6.2. Analisis Hasil Uji Coba Algoritma Genetika dan Pencarian Tabu**

Dari hasil uji coba kombinasi algoritma genetika dan pencarian tabu terhadap penentuan rute Depot Air Minum Banyu Belik yang diujikan dalam parameter yang berbeda-beda didapatkan beberapa rute dengan total jarak yang berbeda.

Berdasarkan hasil dari Tabel 6.9, Tabel 6.10 dan Tabel 6.11 yaitu uji coba menggunakan parameter  $P_c=0.2-0.9$ ,  $P_m$  0.05 dan  $N_{pop}$  100, 200 dan 300 diperoleh total jarak terpendek pada setiap tabel yaitu pada Tabel 6.9 dengan  $P_c = 0.2$ , dengan jarak 1821.5 km; pada Tabel 6.10 dengan  $P_c = 0.2$ , dengan jarak 1838.2 km dan pada Tabel 6.11 dengan  $P_c = 0.8$ , dengan jarak 1679.01 km. Hal ini menunjukkan bahwa jarak terendah berada pada Tabel 6.11 dengan nilai  $P_c = 0.8$  dengan jarak 1679.01 km.

Berdasarkan hasil dari Tabel 6.12, Tabel 6.13 dan Tabel 6.14 yaitu uji coba menggunakan parameter  $P_c$  0.2-0.9,  $P_m$  0.1 dan  $N_{pop}$  100, 200 dan 300 diperoleh total jarak terpendek pada setiap tabel yaitu pada Tabel 6.12 dengan  $P_c = 0.7$ , dengan jarak 1682.2 km; pada Tabel 6.13 dengan  $P_c = 0.5$ , dengan jarak 1690.9 km dan pada Tabel 6.14 dengan  $P_c = 0.8$ , dengan jarak 1723.2 km. Hal ini menunjukkan bahwa jarak terendah berada pada Tabel 6.12 dengan nilai  $P_c = 0.7$  dengan jarak 1682.2 km.

Tabel 6.16 merupakan perbandingan hasil terbaik dari masing-masing uji coba kombinasi algoritma genetika dan pencarian tabu dengan parameter yang berbeda-beda. Didapatkan hasil bahwa jarak terbaik menggunakan kombinasi algoritma genetika dan pencarian tabu adalah pada saat  $P_c = 0.8$ ,  $P_m = 0.05$  dan  $N_{pop}$  300 yaitu dengan total jarak 1679.01 km.

**Tabel 6.28 Total Jarak Terbaik Algoritma GA-TS**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak</b>
0.2	0.05	100	1821.5
0.2	0.05	200	1838.2
0.8	0.05	300	1679.01
0.7	0.1	100	1682.2
0.5	0.1	200	1690.9
0.8	0.1	300	1723.2

### **6.6.3. Perbandingan Hasil Algoritma Genetika dengan Algoritma Genetika dan Pencarian Tabu**

Setelah melakukan analisa jarak terbaik pada masing – masing algoritma yaitu algoritma genetika dan kombinasi GA-TS, maka selanjutnya adalah membandingkan kedua algoritma tersebut untuk mengetahui jarak terbaik yang didapatkan dari dataset Solomon RC105 [5]. Pada tabel 6.29 ditampilkan hasil uji coba terbaik menggunakan algoritma genetika dengan kombinasi algoritma genetika dan pencarian tabu.

**Tabel 6.29 Perbandingan Kedua Algoritma**

<b>Pc</b>	<b>Pm</b>	<b>Npop</b>	<b>Total Jarak</b>
0.5	0.05	300	1793.4 (GA)
0.8	0.05	300	1679.01 (GA-TS)

Menurut tabel 6.29 didapatkan kesimpulan bahwa uji coba menggunakan kombinasi algoritma genetika dengan pencarian tabu lebih baik dengan rute 1-36-48-69-72-75-77-32-36-87-40-14-68-24-56-46-54-22-82-64-27-37-93-55-61-81-70-90-48-60-8-63-10-54-56-38-12-25-51-72-64-84-45-43-83-34-28-5-44-9-59-80-37-81-39-31-84-57-13-3-18-96-64-18-49-29-43-53-56-42-66-46-43-17-12-82-13-20-15-50-92-32-47-

3-65-12-98-29-74-75-99-90-32-13-33-10-58-83-27-47-1 dibandingkan dengan uji coba hanya dengan algoritma genetika dengan rute 1-47-48-69-72-75-77-32-36-87-40-14-68-24-56-46-54-22-82-64-27-37-93-55-61-81-70-90-48-60-8-63-10-54-56-38-12-25-51-72-64-84-45-43-83-34-28-5-44-9-59-80-37-81-39-31-84-57-13-3-18-96-64-18-49-29-43-53-56-42-66-46-43-17-12-82-13-20-15-50-92-32-47-3-65-12-98-29-74-75-99-90-32-13-33-10-58-83-27-86-1. Hasil dari algoritma GA-TS yang telah dibuat sama dengan penelitian sebelumnya Jean Berger, Mohamed Barkaoui yang berjudul “*A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem*”. Persentase tingkat efisiensi hasil hybrid GA-TS dan Algoritma Genetika pada Dataset Solomon RC 105 6.4%

## **6.7. Analisa Hasil Uji Coba Data Perusahaan**

Analisis hasil berisi penjelasan tentang semua hasil output yang didapat dari ujicoba yang dilakukan dengan input parameter yang berbeda-beda.

### **6.7.1. Analisa Hasil Uji Coba Algoritma Genetika**

Dari hasil uji coba algoritma genetika terhadap penentuan rute Depot Air Minum Banyu Belik yang diujikan dalam parameter yang berbeda-beda didapatkan beberapa rute dengan total jarak yang berbeda.

Berdasarkan hasil dari Tabel 6.15, Tabel 6.16 dan Tabel 6.17 yaitu uji coba menggunakan parameter  $P_c=0.2-0.9$ ,  $P_m=0.05$  dan  $N_{pop}=100, 200$  dan  $300$  diperoleh total jarak terpendek pada setiap tabel yaitu pada Tabel 6.15 dengan  $P_c = 0.5$ , dengan jarak 734.5 km; pada Tabel 6.16 dengan  $P_c = 0.8$ , dengan jarak 754 km dan pada Tabel 6.17 dengan  $P_c = 0.7$ , dengan jarak 748.4 km. Hal ini menunjukkan bahwa jarak terendah berada pada Tabel 6.15 dengan nilai  $P_c = 0.5$  dengan jarak 734.5 km.

Berdasarkan hasil dari Tabel 6.18, Tabel 6.19 dan Tabel 6.20 yaitu uji coba menggunakan parameter  $P_c 0.2-0.9$ ,  $P_m 0.1$  dan

$N_{pop}$  100, 200 dan 300 diperoleh total jarak terpendek pada setiap tabel yaitu pada Tabel 6.18 dengan  $P_c = 0.7$ , dengan jarak 731.1 km; pada Tabel 6.19 dengan  $P_c = 0.8$ , dengan jarak 757.6 km dan pada Tabel 6.20 dengan  $P_c = 0.7$ , dengan jarak 737.1 km. Hal ini menunjukkan bahwa jarak terendah berada pada Tabel 6.18 dengan nilai  $P_c = 0.7$  dengan jarak 731.1 km.

Tabel 6.30 merupakan perbandingan hasil terbaik dari masing-masing uji coba algoritma dengan parameter yang berbeda-beda. Didapatkan hasil bahwa jarak terbaik menggunakan algoritma genetika adalah pada saat  $P_c = 0.7$ ,  $P_m = 0.1$  dan  $N_{pop}$  100 yaitu dengan total jarak 731.1 km.

**Tabel 6.30 Hasil Terbaik Dari Masing-Masing Uji Coba Algoritma**

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA
<b>0.5</b>	0.05	100	734.5
<b>0.8</b>	0.05	200	754
<b>0.3</b>	0.1	200	763.7
<b>0.7</b>	0.05	300	748.4
<b>0.7</b>	0.1	100	731.1
<b>0.8</b>	0.1	200	757.6
<b>0.6</b>	0.1	300	737.1

### **6.7.2. Analisis Hasil Uji Coba Kombinasi Algoritma Genetika dan Pencarian Tabu**

Dari hasil uji coba kombinasi algoritma genetika dan pencarian tabu terhadap penentuan rute Depot Air Minum Banyu Belik

yang diujikan dalam parameter yang berbeda-beda didapatkan beberapa rute dengan total jarak yang berbeda.

Berdasarkan hasil dari Tabel 6.21, Tabel 6.22 dan Tabel 6.23 yaitu uji coba menggunakan parameter  $P_c=0.2-0.9$ ,  $P_m$  0.05 dan  $N_{pop}$  100, 200 dan 300 diperoleh total jarak terpendek pada setiap tabel yaitu pada Tabel 6.21 dengan  $P_c = 0.5$ , dengan jarak 692.7 km; pada Tabel 6.22 dengan  $P_c = 0.7$ , dengan jarak 693.2 km dan pada Tabel 6.23 dengan  $P_c = 0.5$ , dengan jarak 688.9km. Hal ini menunjukkan bahwa jarak terendah berada pada Tabel 6.23 dengan nilai  $P_c = 0.5$  dengan jarak 688.9 km.

Berdasarkan hasil dari Tabel 6.24, Tabel 6.25 dan Tabel 6.26 yaitu uji coba menggunakan parameter  $P_c$  0.2-0.9,  $P_m$  0.1 dan  $N_{pop}$  100, 200 dan 300 diperoleh total jarak terpendek pada setiap tabel yaitu pada Tabel 6.24 dengan  $P_c = 0.7$ , dengan jarak 688.9 km; pada Tabel 6.25 dengan  $P_c = 0.5$ , dengan jarak 688.8 km dan pada Tabel 6.26 dengan  $P_c = 0.8$ , dengan jarak 695.4 km. Hal ini menunjukkan bahwa jarak terendah berada pada Tabel 6.25 dengan nilai  $P_c = 0.8$  dengan jarak 688.8 km.

Tabel 6.31 merupakan perbandingan hasil terbaik dari masing-masing uji coba kombinasi algoritma genetika dan pencarian tabu dengan parameter yang berbeda-beda. Didapatkan hasil bahwa jarak terbaik menggunakan kombinasi algoritma genetika dan pencarian tabu adalah pada saat  $P_c = 0.8$ ,  $P_m = 0.1$  dan  $N_{pop}$  200 yaitu dengan total jarak 688.8 km.

**Tabel 6.31 Total Jarak Terbaik Algoritma GA-TS**

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA - TS
<b>0.5</b>	0.05	100	692.7

$P_c$	$P_m$	$N_{pop}$	Total Jarak GA - TS
<b>0.7</b>	0.05	200	693.2
<b>0.2</b>	0.05	300	693.1
<b>0.7</b>	0.1	100	688.9
<b>0.8</b>	0.1	200	688.8
<b>0.8</b>	0.1	300	695.4

Berdasarkan uji coba yang dilakukan, analisa dari uji coba yaitu semakin besar nilai  $P_c$ , maka menghasilkan hasil yang lebih baik. Sedangkan  $P_m$  tidak memberikan hasil yang sensitif terhadap hasil. Sedangkan analisa pada populasi, semakin besar jumlah populasi semakin baik hasilnya.

### 6.7.3. Perbandingan Hasil Algoritma Genetika dengan Algoritma Genetika dan Pencarian Tabu

Setelah melakukan analisa jarak terbaik pada masing – masing algoritma genetika dan kombinasi GA-TS, maka selanjutnya adalah membandingkan kedua algoritma tersebut untuk mengetahui jarak terbaik yang didapatkan yang nantinya akan menjadi pertimbangan pihak Banyu Belik dalam melakukan pengiriman galon kepada pelanggan. Pada tabel 6.32 ditampilkan hasil uji coba terbaik menggunakan algoritma genetika dengan kombinasi algoritma genetika dan pencarian tabu.

**Tabel 6.32 Perbandingan Kedua Algoritma**

$P_c$	$P_m$	$N_{pop}$	Total Jarak Terbaik
<b>0.7</b>	0.1	100	731.1 (GA)



0.5	0.1	300	688.8 (GA-TS)
-----	-----	-----	---------------

Menurut tabel 6.32 didapatkan kesimpulan bahwa uji coba menggunakan kombinasi algoritma genetika dengan rute 1-21-7-17-19-5-6-9-15-4-3-12-14-11-8-16-10-2-18-13-20-1 dibandingkan dengan pencarian tabu dengan rute 1-2-7-20-19-5-6-9-14-4-3-12-15-11-8-16-10-21-18-13-17-1 menghasilkan hasil yang lebih baik dibandingkan dengan uji coba hanya dengan algoritma genetika. Persentase tingkat efisiensi hasil *hybrid* GA-TS dibandingkan dengan menggunakan Algoritma Genetika saja pada studi kasus DAMIU Banyu Belik 10.7%. Berdasarkan hasil uji coba dengan parameter Pc maka semakin besar nilai Pc maka menghasilkan hasil output yang lebih baik. Dan berdasarkan parameter Npop, semakin banyak nilai populasi maka menghasilkan kemungkinan yang lebih banyak dan dapat menghasilkan hasil yang lebih minimum.

## 6.8. Validasi

Validasi output dilakukan untuk mengecek kebenaran suatu output algoritma. Dalam tugas akhir ini validasi output dilakukan menggunakan software sederhana yaitu Microsoft Excel seperti ditunjukkan pada tabel 6.33-6.42. Tiap rute yang dihasilkan dan muatan tiap kendaraan akan dihitung secara manual dan dicek apakah output yang dihasilkan oleh GA-TS benar-benar valid dan sesuai dengan hasil pada perhitungan manual.

Tabel 6.33 Validasi Rute Kendaraan Pertama Hari ke-1 Manual

Kendaraan	1						
Hari Ke-	1						
Kapasitas	90						
Route	1	ke	2	ke	7	ke	1
Delivery & Pick Up			45		45		
Kumulatif Delivery			45		90		
Kumulatif Pick-Up			45		90		
Muatan yang dibawa dari node l ke j			90		90		
Jarak antara node customer		18		19.9		26.7	
Kumulatif Jarak		18		37.9		64.6	
Total Jarak Hasil Algoritma	64.6						

1                    2                    7                    1

jarak\_tempuh =

64.6000

Gambar 6.1 Rute Kendaraan Output Algoritma 1

Tabel 6.34 Validasi Rute Kendaraan Pertama Hari ke-2 Manual

Kendaraan	1						
Hari Ke-	2						
Kapasitas	90						
Route	1	ke	5	ke	6	ke	1
Delivery & Pick Up			45		45		
Kumulatif Delivery			45		90		
Kumulatif Pick-Up			45		90		
Muatan yang dibawa dari node l ke j			90		90		
Jarak antara node customer		28.5		9.8		33.4	
Kumulatif Jarak		28.5		38.3		71.7	
Total Jarak Hasil Algoritma	71.7						

Rute\_Akhir =

1      5      6      1

jarak\_tempuh =

71.7000

**Gambar 6.2 Rute Kendaraan Output Algoritma 2**

**Tabel 6.35 Validasi Rute Kendaraan Pertama Hari ke-3 Manual**

Kendaraan	1						
Hari Ke-	3						
Kapasitas	90						
Route	1	ke	4	ke	3	ke	1
Delivery & Pick Up			30		35		
Kumulatif Delivery			30		65		
Kumulatif Pick-Up			30		65		
Muatan yang dibawa dari node i ke j			65		65		
Jarak antara node customer		8		14.6		11.2	
Kumulatif Jarak		8		22.6		33.8	
Total Jarak Hasil Algoritma	33.8						

Rute\_Akhir =  
1        4        3        1

jarak\_tempuh =  
33.8000

Gambar 6.3 Rute Kendaraan Output Algoritma 3

Tabel 6.36 Validasi Rute Kendaraan Pertama Hari ke-4 Manual

Kendaraan	1						
Hari Ke-	4						
Kapasitas	90						
Route	1	ke	11	ke	8	ke	1
Delivery & Pick Up			30		50		
Kumulatif Delivery			30		80		
Kumulatif Pick-Up			30		80		
Muatan yang dibawa dari node i ke j			80		80		
Jarak antara node customer		30.2		14.1		42.2	
Kumulatif Jarak		30.2		44.3		86.5	
Total Jarak Hasil Algoritma	86.5						

Rute\_Akhir =  
1        11        8        1

jarak\_tempuh =  
86.5000

Gambar 6.4 Rute Kendaraan Output Algoritma 4

**Tabel 6.37 Validasi Rute Kendaraan Pertama Hari ke-5 Manual**

Kendaraan	1								
Hari Ke-	5								
Kapasitas	90								
Route	1	ke	21	ke	18	ke	13	ke	1
Delivery & Pick Up			30		35		25		
Kumulatif Delivery			30		65		90		
Kumulatif Pick-Up			30		65		90		
Muatan yang dibawa dari node i ke j			90		90		90		
Jarak antara node customer		55.9		2.3		46.3		8.8	
Kumulatif Jarak		55.9		58.2		104.5		113.3	
Total Jarak Hasil Algoritma	113.3								

`Rute_Akhir =`

`1      21      18      13      1`

`jarak_tempuh =`

`113.3000`

**Gambar 6.5 Rute Kendaraan Output Algoritma 5****Tabel 6.38 Validasi Rute Kendaraan Kedua Hari ke- 1 Manual**

Kendaraan	2							
Hari Ke-	1							
Kapasitas	90							
Route	1	ke	20	ke	19	ke	1	
Delivery & Pick Up			30		45			
Kumulatif Delivery			30		75			
Kumulatif Pick-Up			30		75			
Muatan yang dibawa dari node i ke j			75		75			
Jarak antara node customer		45.2		9.1		54.2		
Kumulatif Jarak		45.2		54.3		108.5		
Total Jarak Hasil Algoritma	108.5							

Rute\_Akhir =  
1      20      19      1

jarak\_tempuh =  
108.5000

Gambar 6.6 Rute Kendaraan Output Algoritma 6

Tabel 6.39 Validasi Rute Kendaraan Kedua Hari ke-2 Manual

Kendaraan	2						
Hari Ke-	2						
Kapasitas	90						
Route	1	ke	9	ke	14	ke	1
Delivery & Pick Up			45		40		
Kumulatif Delivery			45		85		
Kumulatif Pick-Up			45		85		
Muatan yang dibawa dari node I ke j			85		85		
Jarak antara node customer		19.5		12.9		43.5	
Kumulatif Jarak		19.5		32.4		75.9	
Total Jarak Hasil Algoritma	75.9						

Rute\_Akhir =  
1      9      14      1

jarak\_tempuh =  
75.9000

Gambar 6.7 Rute Kendaraan Output Algoritma 7

**Tabel 6.40 Validasi Rute Kendaraan Kedua Hari ke-3 Manual**

Kendaraan	2						
Hari Ke-	3						
Kapasitas	90						
Route	1	ke	12	ke	15	ke	1
Delivery & Pick Up			45		40		
Kumulatif Delivery			45		85		
Kumulatif Pick-Up			45		85		
Muatan yang dibawa dari node l ke j			85		85		
Jarak antara node customer		11.6		6.3		6.4	
Kumulatif Jarak		11.6		17.9		24.3	
Total Jarak Hasil Algoritma	24.3						

1      12      15      1

jarak\_tempuh =

24.3000

**Gambar 6.8 Rute Kendaraan Output Algoritma 8****Tabel 6.41 Validasi Rute Kendaraan Kedua Hari ke-4 Manual**

Kendaraan	2						
Hari Ke-	4						
Kapasitas	90						
Route	1	ke	16	ke	10	ke	1
Delivery & Pick Up			50		15		
Kumulatif Delivery			50		15		
Kumulatif Pick-Up			50		15		
Muatan yang dibawa dari node l ke j			65		65		
Jarak antara node customer		21.1		4.8		14.7	
Kumulatif Jarak		21.1		25.9		40.6	
Total Jarak Hasil Algoritma	40.6						

Rute\_Akhir =  
1      16      10      1  
  
jarak\_tempuh =  
40.6000

Gambar 6.9 Rute Kendaraan Output Algoritma 9

Tabel 6.42 Validasi Rute Kendaraan Kedua Hari ke-5 Manual

Kendaraan	2				
Hari Ke-	5				
Kapasitas	90				
Route	1	ke	17	ke	1
Delivery & Pick Up			60		60
Kumulatif Delivery			60		60
Kumulatif Pick-Up			60		60
Muatan yang dibawa dari node i ke j			60		60
Jarak antara node customer		34.8		34.8	
Kumulatif Jarak		34.8		69.6	
Total Jarak Hasil Algoritma	69.6				

Rute\_Akhir =  
1      17      1  
  
jarak\_tempuh =  
69.6000

Gambar 6.10 Rute Kendaraan Output Algoritma 10



Setelah dibandingkan, solusi yang dihasilkan adalah valid, karena perhitungan manual menggunakan Microsoft Excel dari keseluruhan elemen seperti jarak dan muatan yang dihasilkan oleh Algoritma GA-TS sama.

### 6.9. Perbandingan Rute Hasil GA-TS dengan Rute Awal

Setelah output yang dihasilkan menunjukkan hasil yang valid, selanjutnya data rute tersebut dibandingkan dengan rute awal yang ditentukan oleh Banyu Balik. Membandingkan tersebut mempunyai tujuan, apakah data rute terbaik yang dihasilkan oleh algoritma GA-TS apakah lebih baik atau mungkin lebih buruk dari rute awal yang ditentukan oleh pihak Banyu Belik. Rute awal Banyu belik dapat dilihat pada tabel 6.43 dan 6.44 dan untuk perbandingan rute yang dilewati dapat dilihat pada gambar 6.1 dan 6.2.

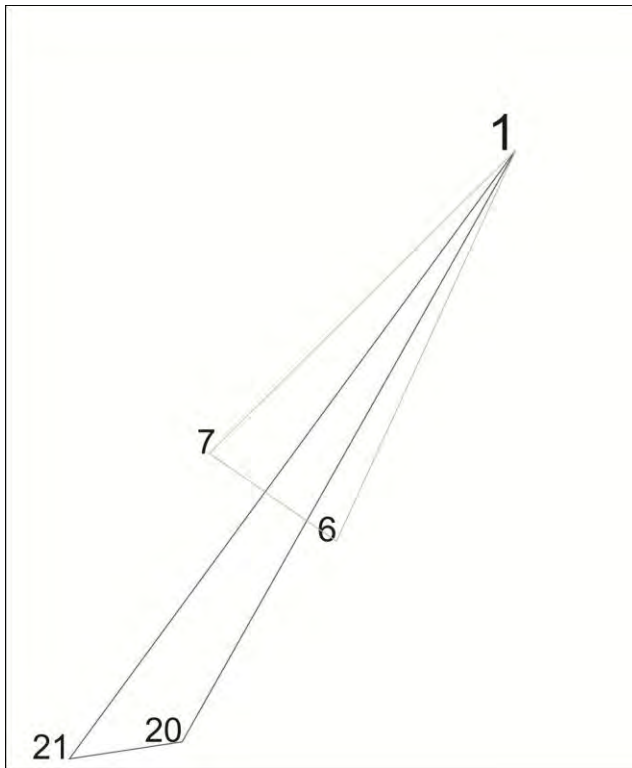
**Tabel 6.43 Rute Awal Kendaraan Pertama**

Kendaraan 1						
Hari ke 1	Rute	1	2	3	1	
	Jarak		18	13.1	11.2	42.3
Hari ke 2	Rute	1	4	16	1	
	Jarak		8	26.2	21.1	55.3
Hari ke 3	Rute	1	5	15	1	
	Jarak		28.5	26.5	6.4	61.4
Hari ke 4	Rute	1	6	7	1	
	Jarak		33.4	14.3	26.7	74.4
Hari ke 5	Rute	1	18	19	1	
	Jarak		54	1	54.1	109.1

**Tabel 6.44 Rute Awal Kendaraan Kedua**

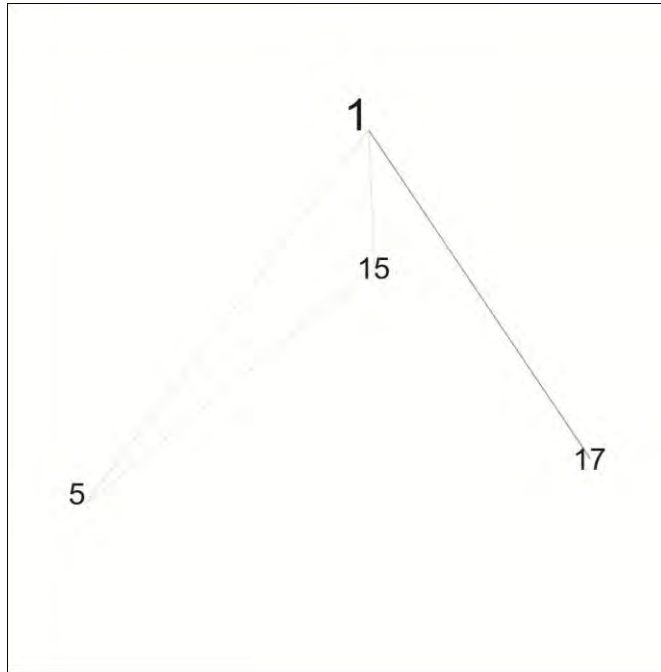
<b>Kendaraan 2</b>							
Hari ke 1	Rute	1	8	11	1		
	Jarak		42.2	14.1	30.2		86.5
Hari ke 2	Rute	1	10	13	14	1	
	Jarak		14.7	8.3	48	45.3	116.3
Hari ke 3	Rute	1	17	1			
	Jarak		34.8	34.8			69.6
Hari ke 4	Rute	1	20	21	1		
	Jarak		45.2	14.5	55.9		115.6
Hari ke 5	Rute	1	9	12	1		
	Jarak		19.5	10	11.6		41.1





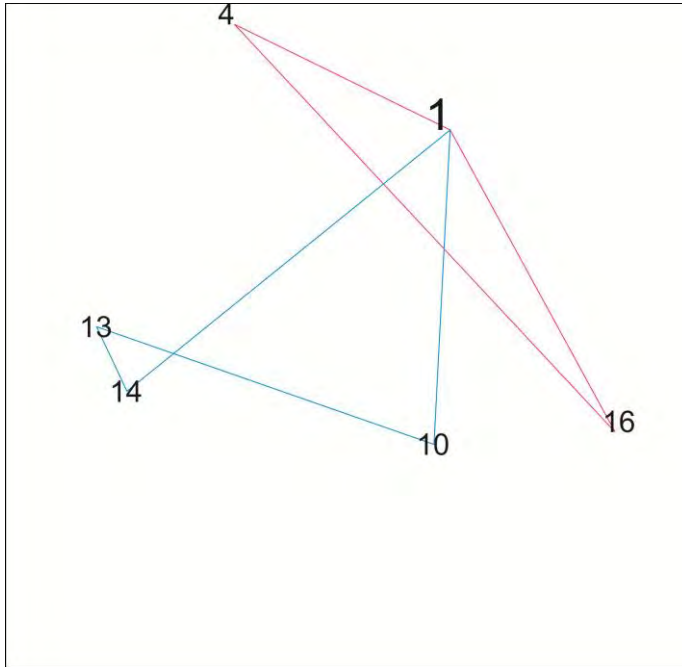
**Gambar 6.12 Rute Hari Pertama**

Pada gambar 6.12 dengan warna biru dengan rute 1-7-6-1 merupakan kendaraan pertama pada hari pertama sedangkan dengan warna ungu dengan rute 1-21-20-1 merupakan kendaraan kedua pada hari pertama.



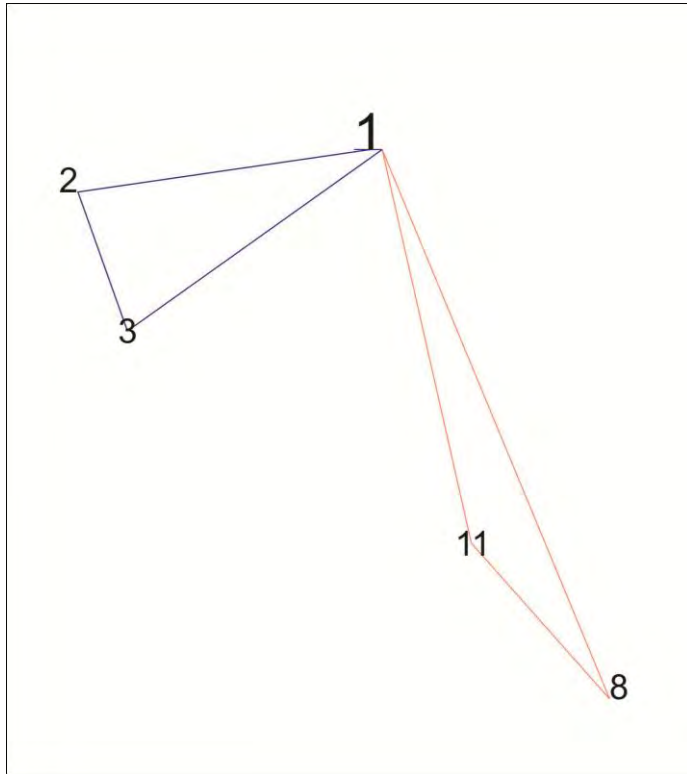
**Gambar 6.13 Rute Hari Kedua**

Pada gambar 6.13 dengan warna biru dengan rute 1-5-15-1 merupakan kendaraan pertama pada hari kedua sedangkan dengan warna ungu dengan rute 1-17-1 merupakan kendaraan kedua pada hari kedua.



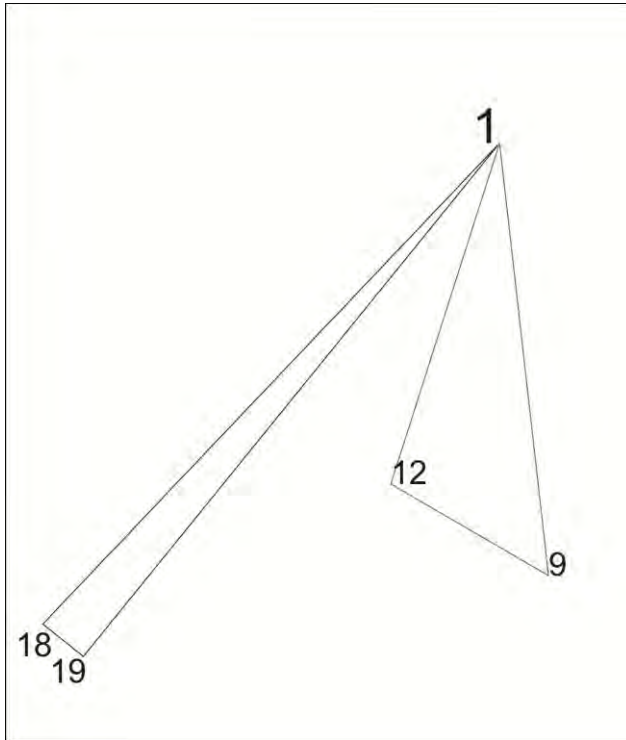
**Gambar 6.14 Rute Hari Ketiga**

Pada gambar 6.14 dengan warna merah muda dengan rute 1-4-16-1 merupakan kendaraan pertama pada hari ketiga sedangkan dengan warna biru dengan rute 1-10-13-14-1 merupakan kendaraan kedua pada hari ketiga.



**Gambar 6.15 Rute Hari Ke-Empat**

Pada gambar 6.15 dengan warna merah muda dengan rute 1-11-8-1 merupakan kendaraan pertama pada hari ke-empat sedangkan dengan warna ungu dengan rute 1-2-3-1 merupakan kendaraan kedua pada hari ke-empat.



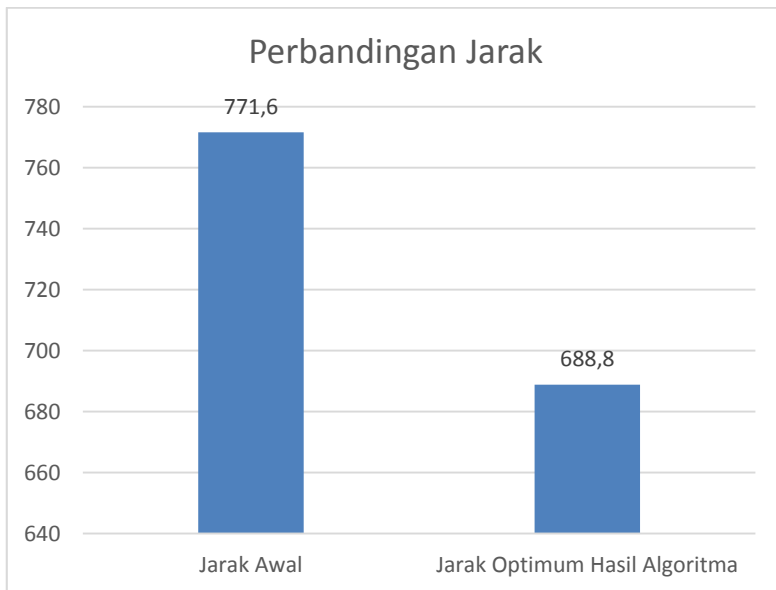
**Gambar 6.16 Rute Hari Ke-Lima**

Pada gambar 6.16 dengan warna biru dengan rute 1-12-9-1 merupakan kendaraan pertama pada hari ke-lima sedangkan dengan warna ungu dengan rute 1-18-19-1 merupakan kendaraan kedua pada hari ke-lima.





Total jarak dari rute awal yang dimiliki Banyu Belik adalah 771.6 km. Bila dibandingkan dengan rute yang dihasilkan oleh algoritma GA-TS yaitu 688.8 km, maka dapat disimpulkan bahwa rute yang dihasilkan oleh kombinasi algoritma genetika dan pencarian tabu lebih pendek dibandingkan dengan rute awal Depot Banyu Belik dengan selisih 82.8 km dengan persentase efisiensi 10.7%.



**Gambar 6.18 Grafik Perbandingan Jarak**

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Bab ini berisikan kesimpulan yang didapat dari seluruh proses pengerjaan tugas akhir dan juga saran perbaikan untuk penelitian kedepannya untuk dapat dikembangkan dari tugas akhir ini.

#### **7.1. Kesimpulan**

Adapun beberapa hal yang dapat disimpulkan dalam pengerjaan Tugas Akhir ini sebagai berikut:

1. Pengoptimasian penentuan rute pengiriman barang dilakukan untuk memperoleh keuntungan yang optimal untuk pihak pengirim, karena dengan jarak terpendek maka pengeluaran yang dikeluarkan untuk bagian distribusi lebih sedikit.
2. Pemilihan jumlah populasi dan nilai probabilitas pindah silang memiliki peran penting dalam optimasi algoritma genetika.
3. Pemilihan jumlah maksimum iterasi dan populasi tabu memiliki peran penting dalam optimasi algoritma pencarian tabu.
4. Jumlah populasi ( $N_{pop}$ ) yang dapat memberikan hasil paling optimal pada permasalahan *Vehicle Routing Problem with Pick-Up and Delivery* adalah 300.
5. Nilai probabilitas pindah silang ( $P_c$ ) yang dapat memberikan hasil paling optimal pada permasalahan *Vehicle Routing Problem with Pick-Up and Delivery* adalah 0.8.
6. Mengkombinasikan algoritma genetika dengan pencarian tabu memberikan hasil yang paling optimal.
7. Rute optimal yang dihasilkan oleh kombinasi GA-TS adalah 1-2-7-20-19-5-6-9-14-4-3-12-15-11-8-16-10-21-18-13-17-1.
8. Total jarak dari rute awal yang dimiliki Banyu Belik adalah 771.6 km. Total jarak yang dihasilkan GA-TS

adalah 688.8 km. Penentuan rute menggunakan kombinasi algoritma genetika dan algoritma pencarian tabu menghasilkan rute yang lebih optimal daripada rute yang telah ditentukan Depot Air Minum Isi Ulang Banyu Belik.

9. Lama perjalanan total adalah 17 jam, dengan membutuhkan waktu lima hari dengan dua kendaraan yang telah tersedia oleh Depot Banyu Belik.
10. Efektifitas yang dihasilkan GA-TS pada studi kasus pengiriman barang DAMIU Banyu Belik memberikan prosentase efektifitas sebesar 10,7%

## 7.2. Saran

1. Beberapa variabel yang belum dipertimbangkan dalam penelitian ini menjadi celah untuk disempurnakan pada penelitian selanjutnya, misalnya penambahan kendala time windows, kendala armada yang tersedia atau *customer satisfaction level* dan sebagainya.
2. Perlunya implementasi algoritma metaheuristik lainnya untuk pembandingan, contohnya *Differential Evolution*, *Ant Colony*, *Particle Swarm*, *Simulated Annealing*, *Hill Climbing* dan yang lainnya dengan menggunakan data yang serupa, sehingga diketahui algoritma yang terbaik dalam penyelesaian kasus *Vehicle Routing Problem with Delivery and Pick-Up* di antara algoritma-algoritma tersebut.

## DAFTAR PUSTAKA

- [1] A. S. Tasan and M. Gen, "A Genetic Algorithm Based Approach to Vehicle Routing Problem with Simultaneous Pick-up and Deliveries," *Computer & Industrial Engineering*, vol. 62, pp. 755-761, 2012.
- [2] S. Yu, C. Ding and K. Zhu, "A hybrid GA-TS algorithm for open vehicle routing optimization of coal mines material," *Expert System with Applications*, pp. 10568-10573, 2011.
- [3] C. O. Yang and A. Utamima, "Hybrid Estimation of Distribution Algorithm for solving Single Row Facility Layout Problem," *Computers & Industrial Engineering*, 2013.
- [4] B. Santosa and P. Willy, *Metoda Metaheuristik Konsep dan Implementasi*, Surabaya: Guna Widya, 2011.
- [5] J. Berger and M. Barkaoui, "A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem".
- [6] J. Wu and Z. Lin, "Research on customer segmentation model by clustering," *ACM International Conference Proceeding Series*, p. 113, 2005.
- [7] J.-T. Wei, S.-Y. Ling and H.-H. Wu, "A Review of the Application of RFM Model," *Academic Journal*, 2010.
- [8] H. A. Wakhidah, 7 November 2013. [Online]. Available: <http://www.solopos.com/2013/11/07/industri-farmasi-30-pasar-farmasi-masih-dikuasai-asing-463200>.
- [9] B. Stone, *Successful direct marketing methods*, Lincolnwood: NTC Business Books, 1995.
- [10] J. Stefanowski, *On rough based approaches to induction of decision rules*, Heidelberg, 1998.

- [11] B. Shim, K. Choi and Y. Suh, "CRM strategies for a small-sized online shopping mall based on association rules," *Expert Systems with Application*, pp. 7736-7741, 2012.
- [12] D. Peppers and M. Rogers, *Managing Customer Relationships : second edition*, John Wiley & Sons, Inc, 2011.
- [13] Z. Pawlak, "Rough Sets," *Informational Journal of Computer and Information Sciences*, pp. 341-356, 1982.
- [14] M. B. Pamungkas, February 2013. [Online]. Available: <http://ayahsafa.blogspot.com/2013/02/tugas-medreps-mengelola-area-coverage.html>.
- [15] M. B. Pamungkas, February 2013. [Online]. Available: <http://ayahsafa.blogspot.com/2013/02/tugas-medical-representative-menjaga.html>.
- [16] M. B. Pamungkas, May 2013. [Online]. Available: <http://indonesia-pharmacommunity.blogspot.com/2013/05/peluang-besar-di-industri-farmasi-2014.html>.
- [17] E. Ngai, L. Xlu and D. Chau, "Application Of Data Mining Technique in Customer Reationship Management: A Literature Review and Classification," pp. 2592-2602, 2009.
- [18] D. M. A. B. Joni and V. Nurcahyawati, "Penentuan Jarak Terpendek pada Jalur Distribusi Barang di Pulau Jawa dengan Menggunakan Algoritma Genetika," 2011.
- [19] A. Hughes, *Strategic Database marketing*, Chicago: Probus Publishing Company, 1994.
- [20] S. H. Han, S. X. Lu and S. C. Leung, "Segmentation of telecom customers based on customer value by decision," pp. 3964-3973, 2011.
- [21] J. W. Grzymala-Busse, "Rule Induction," *Fundamenta Informaticae*, 2002.

- [22] C.-H. Cheng and Y.-S. Chen, "Classifying the segmentation of customer value via RFM model and RS theory," *Expert Systems with Application*, pp. 4176-4183, 2009.
- [23] R. Buaton, 15 metode menyelesaikan data mining, sistem pakar dan sistem pendukung keputusan, Medan, 2014.
- [24] Kalbe Farma Tbk, "Materic Public Expose," Jakarta, 2013.
- [25] M. Ashoori and Z. Taheri, "Using Clustering Methods for Identifying Blood Donors Behavior," *5th Iranian Conference on Electrical and Electronics Engineering*, pp. 4055-4058, 2013.
- [26] A. Widhiyasa, "Kajian Algoritma Genetika," *Penyelesaian Traveling Salesman Problem*, pp. 1-11, 2010.
- [27] F. Glover, "Tabu Search Fundamentals and Uses," 1995.
- [28] J. Dethloff, "Vehicle Routing and Reverse Logistics," *The Vehicle Routing Problem with Simultaneous delivery and pick-up*, no. 23, pp. 79-96, 2001.
- [29] G. B. Dantzig and R. H. Ramser, "The truck dispatching problem.," *Management Science.*, vol. 6, pp. 80-91, 1959.
- [30] N. a. E. Optimization, "VRP with Pick-Up and Delivering," *Networking and Emerging Optimization*, 7 1 2013. [Online]. Available: <http://neo.lcc.uma.es>. [Accessed 1 3 2015].
- [31] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithm*, 2nd ed., Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2004.
- [32] E. Turban, R. E. Sharda and D. Delen, *Decision Support and Business Intelligence Systems*, New Jersey: Prentice Hall, 2010.
- [33] Q. Jiajiao, Z. Yong, M. Jianlin and F. Lixia, "A New Coding Method for Genetic Algorithm in Vehicle Routing Problem," *International Conference on Cyber*

*Technology in Automation, Control, and Intelligent Systems*, pp. 201-204, 2012.

- [34] D. Zheng-yu, Y. Dong-yuan and W. Shang, "An Improved Genetic Algorithm for Time Dependent VRP," 2010.
- [35] H. Min, W. Zhuo and H. Linghui, "The Study of Optimizing of Physical Distribution Routing Problem System with Time Windows Based on GA," 2010.
- [36] M. Mitchell, *An Introduction to Genetic Algorithm*, London, England: The MIT Press, 1999.
- [37] K. A. Prabha and R. Saranya, "Refinement of K-Means Clustering Using Genetic Algorithm," *Journal of Computer Applications (JCA)*, vol. IV, no. 2, 2011.
- [38] F. Glover and M. Laguna, "Tabu Search," *Boston: Kluwer Academic Publisher*, 1997.
- [39] F. Glover, "Genetic Algorithms and Tabu Search: Hybrids for Optimization," *Computer and Operation Research*, pp. 111-134, 1995.
- [40] R. Thamilselvan and P. Balasubramanie, "A Genetic Algorithm with a Tabu Search (GTA) for Traveling Salesman Problem," *International Journal of Recent Trends in Engineering*, pp. 607-610, 2009.
- [41] N. Christofides, A. Mingozzi and P. Toth, *Combinatorial Optimization*, Chichester, UK: John Wiley & Sons, 1979.



## A. LAMPIRAN A

### DATA JARAK PERUSAHAAN

**Tabel 8.1 Data Jarak (1)**

<b>Node</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>1</b>	0	18	11.2	8	28.5	33.4	26.7	42.2	19.5	14.7	30.2
<b>2</b>	18	0	14.2	21.5	23	35.2	19.9	48.4	26.8	22	37.5
<b>3</b>	11.20	13.1	0	14.6	19.7	27.7	16.5	38.1	20.5	15.1	27.2
<b>4</b>	8	21.5	14.6	0	33.2	39.7	30	48.4	26.8	22	37.5
<b>5</b>	28.5	23	19.7	33.2	0	9.8	5	35.1	24.7	28.7	31.4
<b>6</b>	33.4	35.2	27.8	39.8	9.9	0	14.4	36.9	30	34	38.5
<b>7</b>	26.7	19.9	16.5	30	5	14.3	0	37	24.5	26.7	31.2
<b>8</b>	42.2	49.2	38.9	49.2	35.1	36.8	37.2	0	22.9	27.9	14.1
<b>9</b>	19.5	25.8	20.5	26.8	24.9	29.8	24.5	22.1	0	5.4	11.1
<b>10</b>	14.7	22.9	15.9	22.9	26.3	31.2	27.5	27.1	5.4	0	16.2
<b>11</b>	30.2	38.3	28	38.3	32.4	38.3	32	14.1	12	17	0

A-2

<b>Node</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>12</b>	11.6	18.5	11.5	18.5	21.8	26.7	21.4	31.7	10	6.1	20.8
<b>13</b>	8.8	16.8	9.9	16.8	22.8	27.8	20.6	34.7	13	8.3	23.8
<b>14</b>	43.5	59.7	52.8	47	46.5	75.5	56.9	66.7	14.3	42	55.3
<b>15</b>	6.4	18.3	11.3	14.6	26.5	31.5	24.3	34.7	13	8.2	23.8
<b>16</b>	21.1	26.2	19.2	26.2	31.8	36.8	31.4	28.2	6.6	4.8	17.3
<b>17</b>	34.8	49.7	42.8	41.7	55.4	60.3	55	40.4	30.1	28.4	34.2
<b>18</b>	54	51.7	48.2	60.2	34	25.2	38.5	47.5	47.7	54.4	49
<b>19</b>	54.2	51.9	48.4	60.4	34.2	25.4	38.7	47.7	47.9	54.6	49.2
<b>20</b>	45.2	35.6	38.8	48.6	24.5	16.1	26.3	51.5	44.6	48.6	53.1
<b>21</b>	55.9	52.4	50.2	62.2	35.9	27.2	40.4	49.4	49.7	56.3	51

**Tabel 8.2 Data Jarak (2)**

<b>Node</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>
<b>1</b>	11.6	8.8	45.3	6.4	21.1	34.8	54	54.1	45.2	55.9
<b>2</b>	19.3	25	55.7	18.5	26.2	49.6	51.7	51.8	35.6	52.4
<b>3</b>	12.3	8	49.4	11.6	37.4	42	48.8	48.9	41.5	50.6

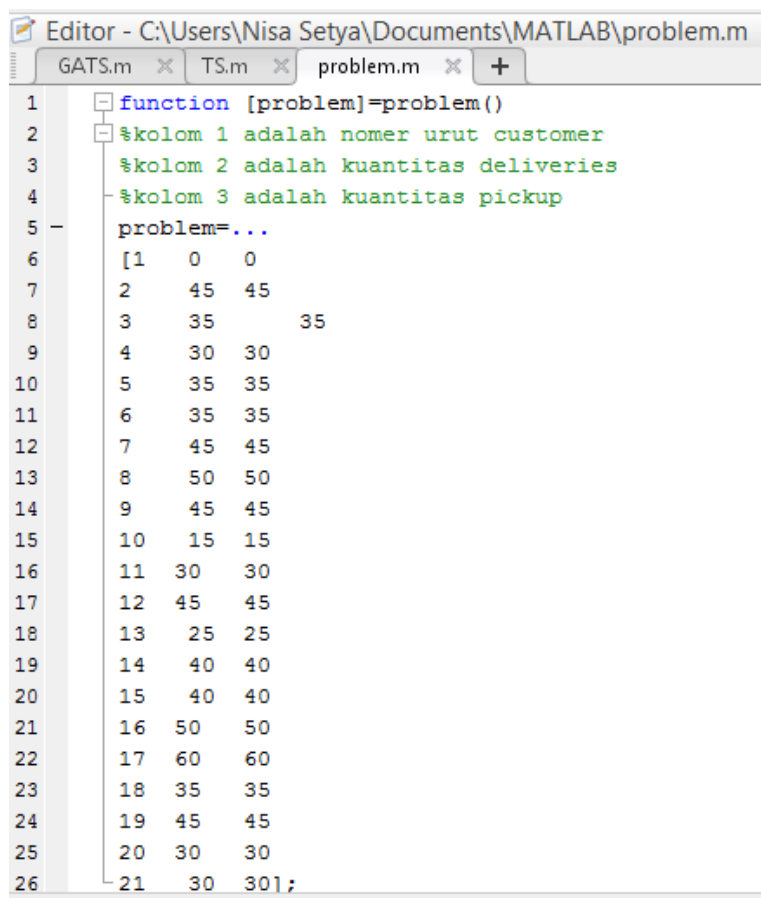
<b>Node</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>
<b>4</b>	19.3	15	47	14.6	32.4	41.6	59.3	59.4	52	61.2
<b>5</b>	21.6	20.7	69.3	26.5	53.7	34.6	34.1	34.2	44.8	50
<b>6</b>	26.9	25.9	74.4	31.7	58.8	60.9	25.3	25.4	23.2	32.5
<b>7</b>	21.4	18.6	56	24.4	53.4	42.5	39.1	39.2	35.5	40.9
<b>8</b>	32.1	35.3	65.9	35.5	49.3	43.4	46	47	48.7	47.9
<b>9</b>	9.6	13.5	12.9	13	28.6	29.6	48.8	48.9	48.8	50.7
<b>10</b>	5.7	9.6	41.1	9.1	26.8	28.9	50.6	50.7	48.6	52.5
<b>11</b>	21.2	24.4	57.6	24.6	39.4	33.1	48.8	48.9	51.4	50.6
<b>12</b>	0	4.4	47	6.3	31.4	33.5	47.2	47.2	43.8	49
<b>13</b>	4.4	0	98.4	6.6	34.4	36.5	47.6	47.6	44.3	49.5
<b>14</b>	47.5	48	0	42.2	47.8	44.9	92.6	92.7	92.5	94.4
<b>15</b>	7.2	6.8	42.3	0	26.7	30.1	51.8	51.9	48.5	53.7
<b>16</b>	9	12.9	37.6	12.4	0	24.1	54	55	53.9	55.9
<b>17</b>	32.5	36.4	45.2	30.6	29.7	0	77.8	77.9	80.1	82.8
<b>18</b>	47.3	46.3	90.6	52.2	75.1	80	0	1	12.9	1.3
<b>19</b>	47.5	46.5	92.6	52.4	77.1	82	1	0	14.9	3.3
<b>20</b>	41.5	40.5	87.5	45.6	73.6	79.1	9.9	9.1	0	14.6

A-4

Node	12	13	14	15	16	17	18	19	20	21
21	49.2	48.2	93.5	54.1	77.9	82.8	2.3	2.4	14.5	0

## B. LAMPIRAN B

### DATA PENGIRIMAN DAN PENGAMBILAN BARANG



The screenshot shows a MATLAB editor window with the title bar "Editor - C:\Users\Nisa Setya\Documents\MATLAB\problem.m". The window contains a function definition for "problem". The function starts with a comment block explaining the columns of the data matrix: column 1 is the customer sequence number, column 2 is the quantity of deliveries, and column 3 is the quantity of pickups. The data matrix is then defined as a 21x3 array. The first row is [1, 0, 0]. The subsequent rows represent 20 customers, each with a unique sequence number (2-21), a delivery quantity, and a pickup quantity. The delivery and pickup quantities are the same for each customer. The function ends with a semicolon.

```
1 function [problem]=problem()  
2 %kolom 1 adalah nomer urut customer  
3 %kolom 2 adalah kuantitas deliveries  
4 %kolom 3 adalah kuantitas pickup  
5 problem=...  
6 [1 0 0  
7 2 45 45  
8 3 35 35  
9 4 30 30  
10 5 35 35  
11 6 35 35  
12 7 45 45  
13 8 50 50  
14 9 45 45  
15 10 15 15  
16 11 30 30  
17 12 45 45  
18 13 25 25  
19 14 40 40  
20 15 40 40  
21 16 50 50  
22 17 60 60  
23 18 35 35  
24 19 45 45  
25 20 30 30  
26 21 30 30];
```

Gambar 8.1 Data Pengiriman dan Pengambilan Barang

## C. LAMPIRAN C

### HASIL RUNNING JARAK TERBAIK

6/30/15 12:08 AM MATLAB Command Window 102 of 105

---

```
1      2      7      1

jarak_tempuh =

    64.6000

Rute_Akhir =

     1     20     19      1

jarak_tempuh =

    108.5000

Rute_Akhir =

     1      5      6      1

jarak_tempuh =

    71.7000

Rute_Akhir =

     1      9     14      1
```

**Gambar 8.2 Hasil Running (1)**

C-2

```
jarak_tempuh =
```

```
75.9000
```

```
Rute_Akhir =
```

```
1    4    3    1
```

```
jarak_tempuh =
```

```
33.8000
```

```
Rute_Akhir =
```

---

6/30/15 12:08 AM MATLAB Command Window

103 of 105

```
1    12   15    1
```

```
jarak_tempuh =
```

```
24.3000
```

---

**Gambar 8.3 Hasil Running (2)**

```
Rute_Akhir =  
    1    11    8    1  
  
jarak_tempuh =  
    86.5000  
  
Rute_Akhir =  
    1    16    10    1  
  
jarak_tempuh =  
    40.6000  
  
Rute_Akhir =  
    1    21    18    13    1  
  
jarak_tempuh =  
    113.3000  
  
Rute_Akhir =  
    1    17    1  
  
jarak_tempuh =  
    69.6000
```

---

**Gambar 8.4 Hasil Running (3)**



```
jarakakhirGA =  
758.8000
```

---

6/30/15 12:08 AM MATLAB Command Window

104 of 105

---

```
total_jarak_GA_TS =  
688.8000
```

```
solusi_akhir =
```

```
Columns 1 through 10
```

```
1    2    7   20   19    5    6    9   14    4
```

```
Columns 11 through 20
```

```
3   12   15   11    8   16   10   21   18   13
```

```
Columns 21 through 22
```

```
17    1
```

**Gambar 8.5 Hasil Running (4)**

## D. LAMPIRAN D

### DATASET SOLOMON RC 105

**Tabel 8.3 Dataset Solomon**

RC105

VEHICLE NUMBER	CAPACITY		
25	200		
CUSTOMER CUST NO.	XCOORD.	YCOORD.	DEMAND
0	40	50	0
1	25	85	20
2	22	75	30
3	22	85	10
4	20	80	40
5	20	85	20
6	18	75	20
7	15	75	20
8	15	80	10
9	10	35	20
10	10	40	30
11	8	40	40
12	8	45	20
13	5	35	10
14	5	45	10
15	2	40	20
16	0	40	20
17	0	45	20
18	44	5	20
19	42	10	40
20	42	15	10
21	40	5	10
22	40	15	40
23	38	5	30
24	38	15	10
25	35	5	20
26	95	30	30
27	95	35	20

## BIODATA PENULIS



Penulis dilahirkan di Sidoarjo pada tanggal 21 Oktober 1994. Penulis merupakan anak kedua dari tiga bersaudara. Penulis telah menempuh pendidikan formal yaitu di SD Muhammadiyah 1 Sidoarjo, SMPN 3 Sidoarjo dan SMAN 1 Sidoarjo. Pada tahun 2011 penulis diterima di jurusan Sistem Informasi – Institut Teknologi Sepuluh Nopember (ITS) dari jalur PKM dan terdaftar dengan NRP

5211100186. Selain kesibukan akademik, penulis yang memiliki hobi *travelling* ini memiliki beberapa pengalaman selama di ITS, yakni baik di tingkat akademik ataupun di tingkat organisasi. Penulis pernah menjadi asisten praktikum, mengikuti berbagai organisasi dan kepanitiaan. Tahun 2012 penulis dipercayakan untuk menjadi staff Himpunan Mahasiswa Sistem Informasi, pada tahun 2013 menjadi kepala divisi di salah satu departemen di Himpunan Mahasiswa Sistem Informasi. Penulis juga pernah melakukan kerja praktik di Telkom Corporate University selama 1,5 bulan di tahun 2014

Pada pengerjaan Tugas Akhir di Jurusan Sistem Informasi ITS, penulis mengambil bidang minat Sistem Pendukung Keputusan dengan topik Optimasi, yakni mengenai penentuan rute kendaraan pada studi kasus VRP-PD. Penulis dapat dihubungi melalui e-mail [nisasetya21@gmail.com](mailto:nisasetya21@gmail.com)

*Halaman ini sengaja dikosongkan*